

P&S Heterogeneous Systems

Programming Heterogeneous Computing
Systems with GPUs and other Accelerators

Dr. Juan Gómez Luna

Prof. Onur Mutlu

ETH Zürich

Fall 2022

3 October 2022

P&S: Heterogeneous Systems (I)

227-0085-51L Projects & Seminars: Programming Heterogeneous Computing Systems with GPUs and other Accelerators

Semester	Autumn Semester 2022
Lecturers	O. Mutlu , J. Gómez Luna
Periodicity	every semester recurring course
Language of instruction	English
Comment	Only for Electrical Engineering and Information Technology BSc. Course can only be registered for once. A repeatedly registration in a later semester is not chargeable.

Courses	Catalogue data	Performance assessment	Learning materials	Groups	Restrictions	Offered in	► Overview
Abstract	The category of "Laboratory Courses, Projects, Seminars" includes courses and laboratories in various formats designed to impart practical knowledge and skills. Moreover, these classes encourage independent experimentation and design, allow for explorative learning and teach the methodology of project work.						
Objective	<p>The increasing difficulty of scaling the performance and efficiency of CPUs every year has created the need for turning computers into heterogeneous systems, i.e., systems composed of multiple types of processors that can suit better different types of workloads or parts of them. More than a decade ago, Graphics Processing Units (GPUs) became general-purpose parallel processors, in order to make their outstanding processing capabilities available to many workloads beyond graphics. GPUs have been a critical key to the recent rise of Machine Learning and Artificial Intelligence, which took unrealistic training times before the use of GPUs. Field-Programmable Gate Arrays (FPGAs) are another example computing device that can deliver impressive benefits in terms of performance and energy efficiency. More specific examples are (1) a plethora of specialized accelerators (e.g., Tensor Processing Units for neural networks), and (2) near-data processing architectures (i.e., placing compute capabilities near or inside memory/storage).</p> <p>Despite the great advances in the adoption of heterogeneous systems in recent years, there are still many challenges to tackle, for example:</p> <ul style="list-style-type: none">- Heterogeneous implementations (using GPUs, FPGAs, TPUs) of modern applications from important fields such as bioinformatics, machine learning, graph processing, medical imaging, personalized medicine, robotics, virtual reality, etc.- Scheduling techniques for heterogeneous systems with different general-purpose processors and accelerators, e.g., kernel offloading, memory scheduling, etc.- Workload characterization and programming tools that enable easier and more efficient use of heterogeneous systems. <p>If you are enthusiastic about working hands-on with different software, hardware, and architecture projects for heterogeneous systems, this is your P&S. You will have the opportunity to program heterogeneous systems with different types of devices (CPUs, GPUs, FPGAs, TPUs), propose algorithmic changes to important applications to better leverage the compute power of heterogeneous systems, understand different workloads and identify the most suitable device for their execution, design optimized scheduling techniques, etc. In general, the goal will be to reach the highest performance reported for a given important application.</p> <p>The course is conducted in English.</p> <p>The course has two main parts:</p> <ul style="list-style-type: none">Weekly lectures on GPU and heterogeneous programming.Hands-on project: Each student develops his/her own project. <p>Course website: https://safari.ethz.ch/projects_and_seminars/doku.php?id=heterogeneous_systems ►</p>						

P&S: Heterogeneous Systems (II)

The increasing difficulty of scaling the performance and efficiency of CPUs every year has created the need for turning computers into heterogeneous systems, i.e., systems composed of multiple types of processors that can suit better different types of workloads or parts of them. More than a decade ago, Graphics Processing Units (GPUs) became general-purpose parallel processors, in order to make their outstanding processing capabilities available to many workloads beyond graphics. GPUs have been critical key to the recent rise of Machine Learning and Artificial Intelligence, which took unrealistic training times before the use of GPUs. Field-Programmable Gate Arrays (FPGAs) are another example computing device that can deliver impressive benefits in terms of performance and energy efficiency. More specific examples are (1) a plethora of specialized accelerators (e.g., Tensor Processing Units for neural networks), and (2) near-data processing architectures (i.e., placing compute capabilities near or inside memory/storage).

Despite the great advances in the adoption of heterogeneous systems in recent years, there are still many challenges to tackle, for example:

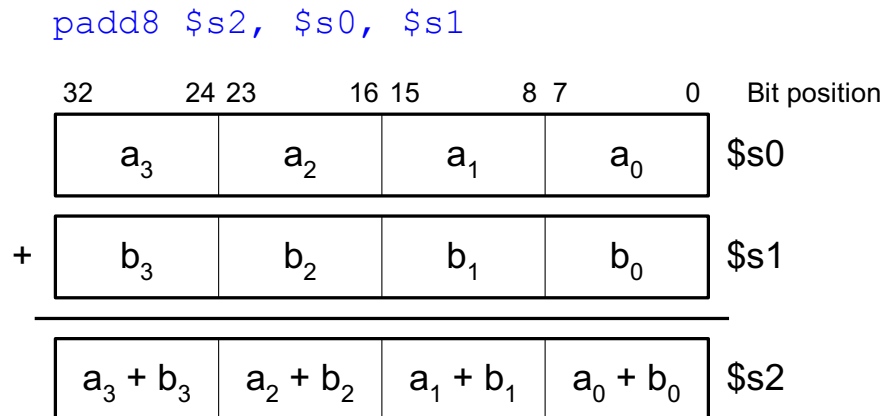
- Heterogeneous implementations (using GPUs, FPGAs, TPUs) of modern applications from important fields such as bioinformatics, machine learning, graph processing, medical imaging, personalized medicine, robotics, virtual reality, etc.
- Scheduling techniques for heterogeneous systems with different general-purpose processors and accelerators, e.g., kernel offloading, memory scheduling, etc.
- Workload characterization and programming tools that enable easier and more efficient use of heterogeneous systems.

Flynn's Taxonomy of Computers

- Mike Flynn, “[Very High-Speed Computing Systems](#),” Proc. of IEEE, 1966
- **SISD**: Single instruction operates on single data element
- **SIMD**: Single instruction operates on multiple data elements
 - Array processor
 - Vector processor
- **MISD**: Multiple instructions operate on single data element
 - Closest form: systolic array processor, streaming processor
- **MIMD**: Multiple instructions operate on multiple data elements (multiple instruction streams)
 - Multiprocessor
 - Multithreaded processor

SIMD ISA Extensions

- Single Instruction Multiple Data (SIMD) extension instructions
 - Single instruction acts on multiple pieces of data at once
 - Common application: graphics
 - Perform short arithmetic operations (also called *packed arithmetic*)
- For example: add four 8-bit numbers
- Must modify ALU to eliminate carries between 8-bit values



Intel Pentium MMX Operations

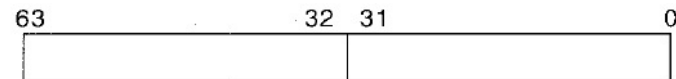
- Idea: One instruction operates on multiple data elements **simultaneously**
 - ❑ *A la* array processing (yet much more limited)
 - ❑ Designed with multimedia (graphics) operations in mind



(a)



(b)



(c)



(d)

No VLEN register

Opcode determines data type:

8 8-bit bytes

4 16-bit words

2 32-bit doublewords

1 64-bit quadword

Stride is always equal to 1.

Peleg and Weiser, “**MMX Technology Extension to the Intel Architecture**,”
IEEE Micro, 1996.

Figure 1. MMX technology data types: packed byte (a), packed word (b), packed doubleword (c), and quadword (d).

MMX Example: Image Overlaying (I)

- Goal: Overlay the human in image x on top of the background in image y

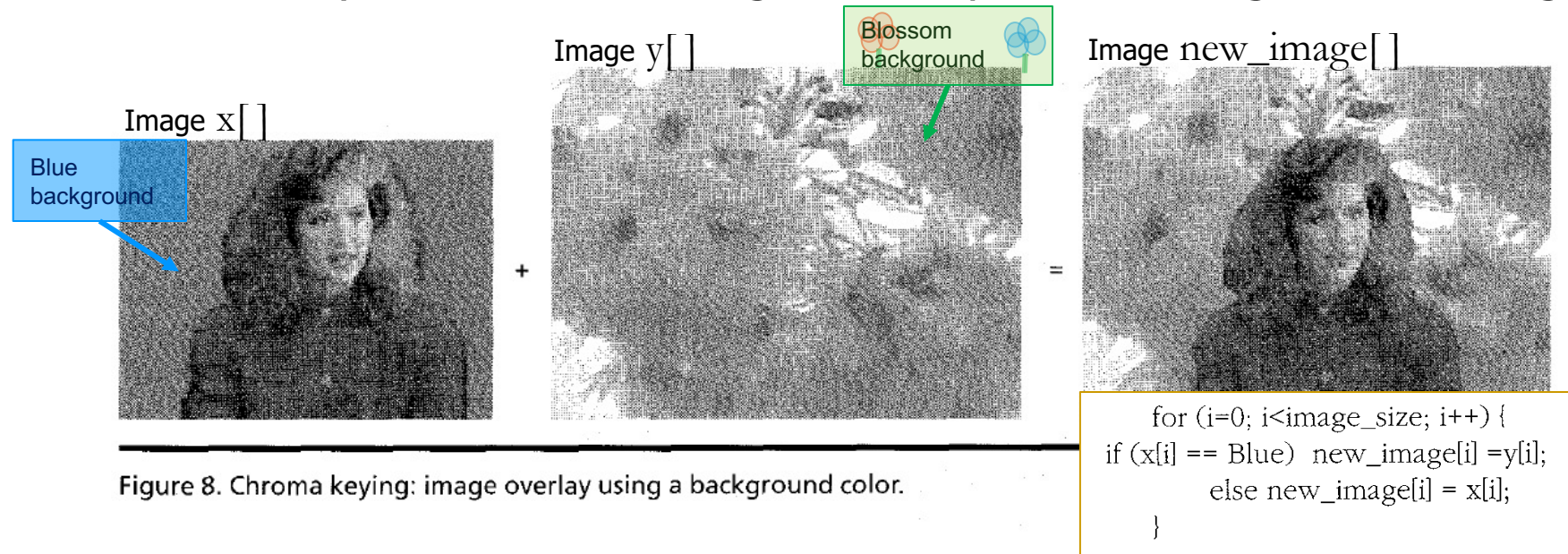


Figure 8. Chroma keying: image overlay using a background color.

PCMPEQB MM1, MM3

MM1	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
MM3	X7!=blue	X6!=blue	X5=blue	X4=blue	X3!=blue	X2!=blue	X1=blue	X0=blue
MM1	0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF



Bitmask

Figure 9. Generating the selection bit mask.

MMX Example: Image Overlaying (II)

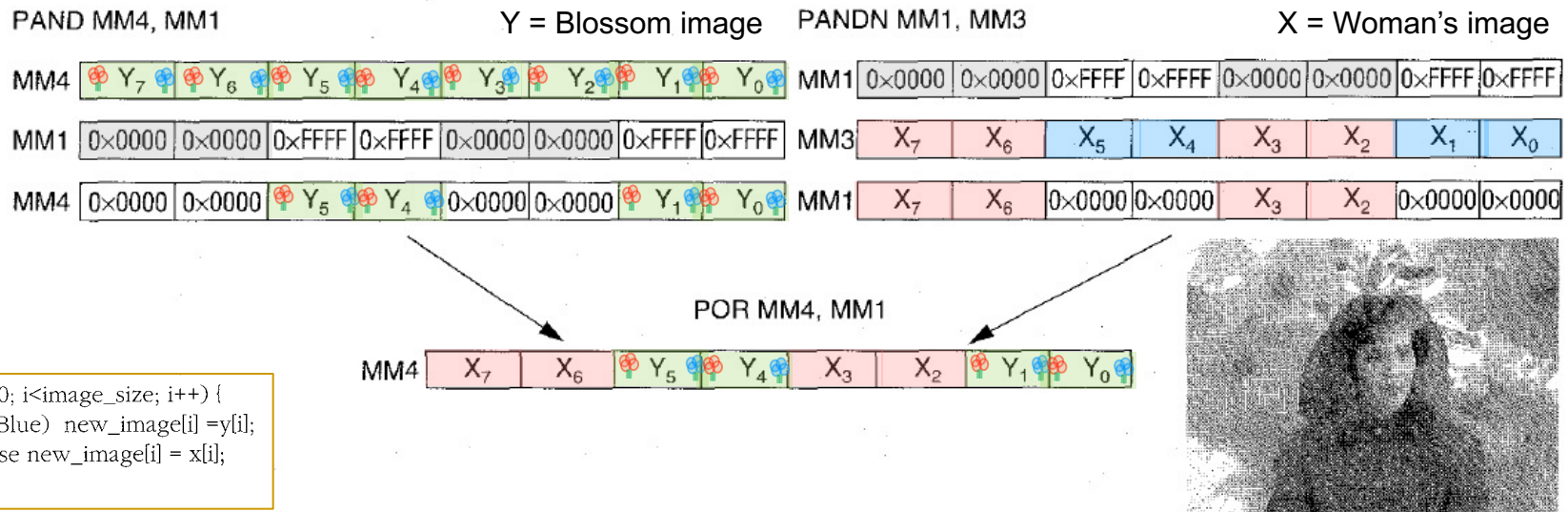


Figure 10. Using the mask with logical MMX instructions to perform a conditional select.

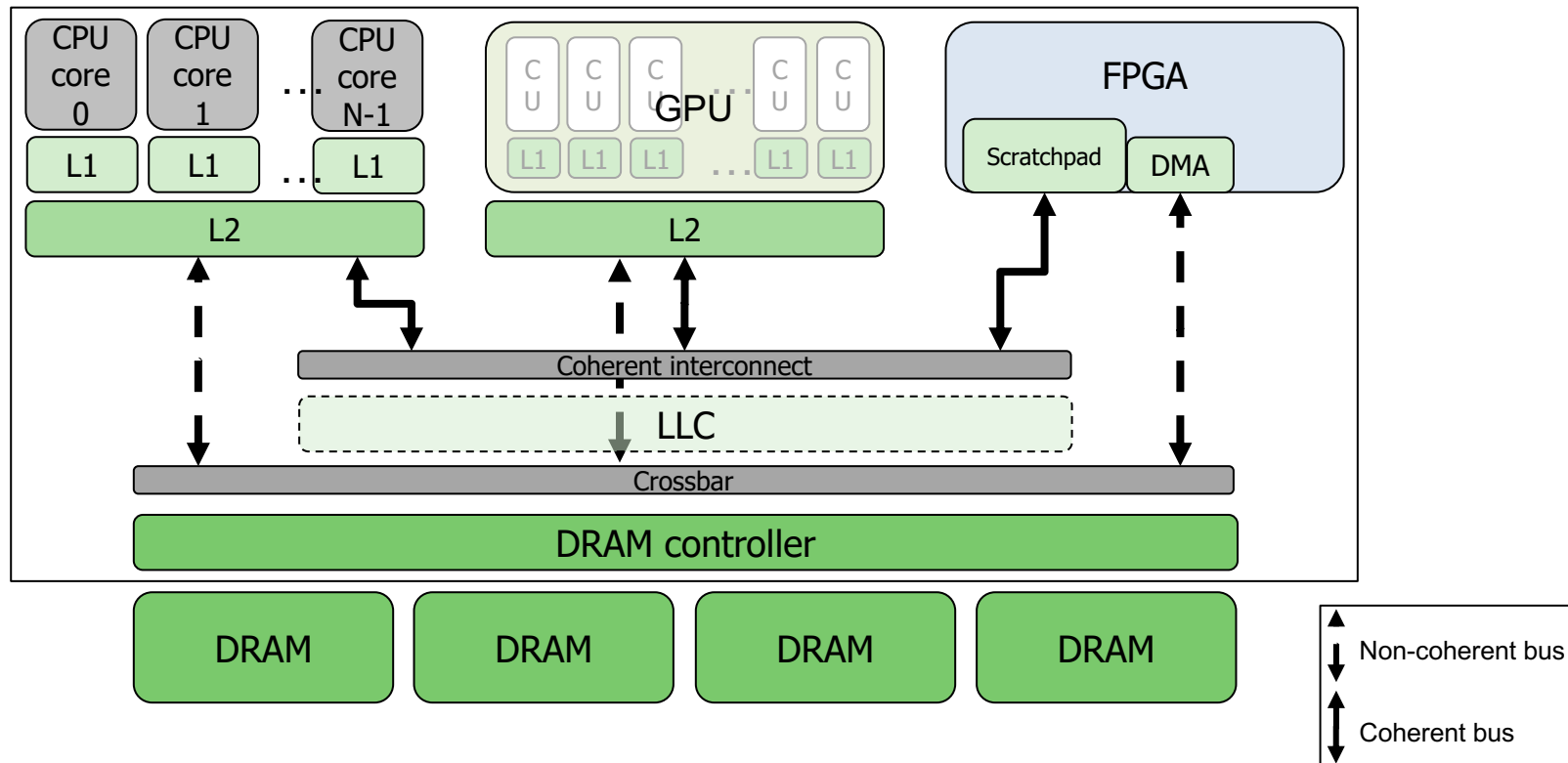
```

Movq    mm3, mem1    /* Load eight pixels from
                      woman's image
Movq    mm4, mem2    /* Load eight pixels from the
                      blossom image
Pcmpeqb mm1, mm3
Pand    mm4, mm1
PANDN   mm1, mm3
POR     mm4, mm1
    
```

Figure 11. MMX code sequence for performing a conditional select.

Heterogeneous Computing Systems

- The end of Moore's law created **the need for heterogeneous systems**
 - More **suitable devices** for each type of workload
 - Increased **performance and energy efficiency**

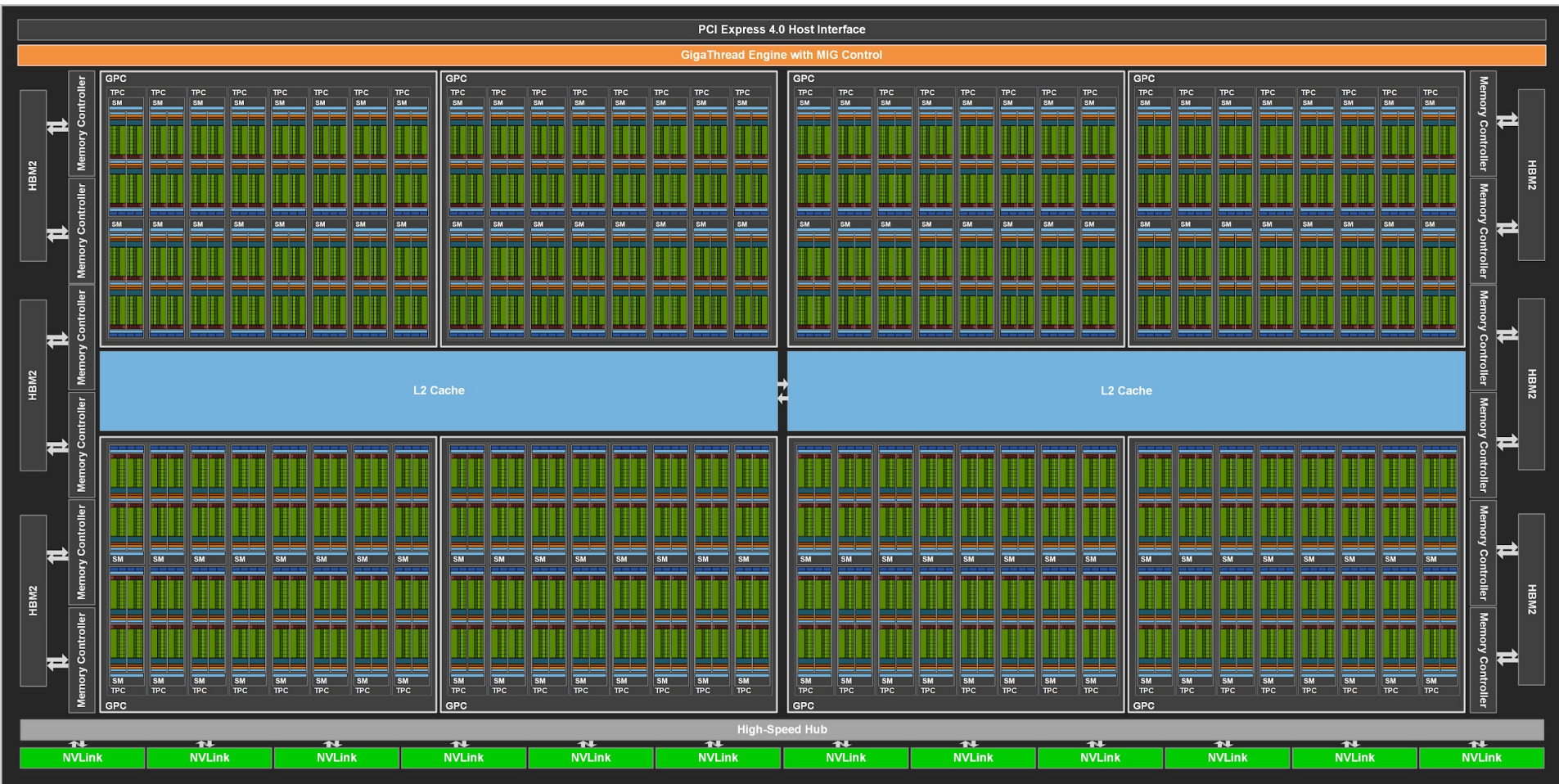


Goals of this P&S Course

P&S Heterogeneous Systems: Contents

- We will introduce the **need for heterogeneity** in current computing systems, in order to achieve high performance and energy efficiency
- You will get familiar with some of the **different heterogeneous devices** that are available in computing systems
- You will learn **GPU programming** with CUDA/OpenCL
- You will learn **workload distribution and parallelization strategies** that leverage heterogeneous devices
- You will **work hands-on**: analyzing workloads, programming heterogeneous architectures, proposing scheduling/offloading mechanisms, etc.

NVIDIA A100 (2020)



<https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/>

108 cores on the A100
(Up to 128 cores in the full-blown chip)

40MB L2 cache

NVIDIA A100 Core

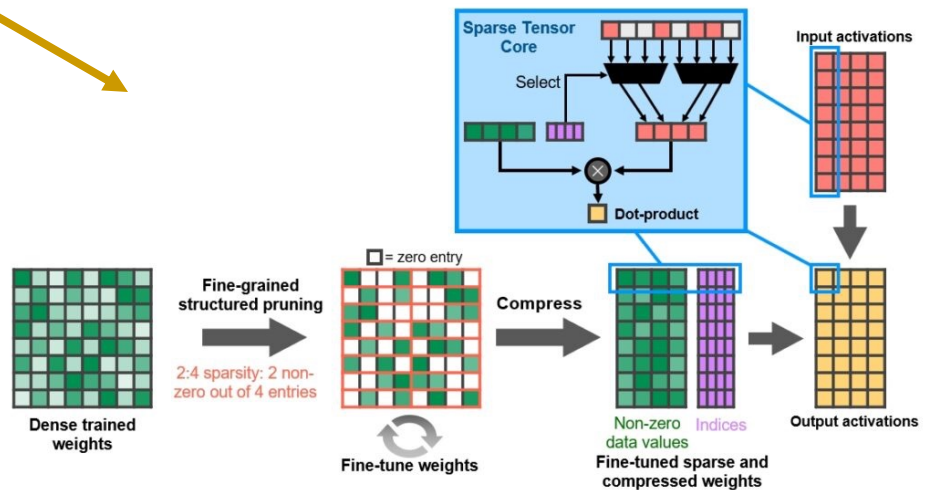


GPU compute throughput:

19.5 TFLOPS Single Precision

9.7 TFLOPS Double Precision

312 TFLOPS for Deep Learning (Tensor cores)



NVIDIA H100 Block Diagram



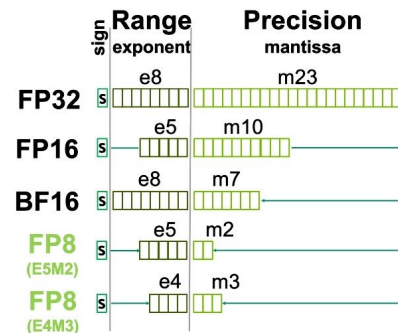
<https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/>

144 cores on the full GH100
60MB L2 cache

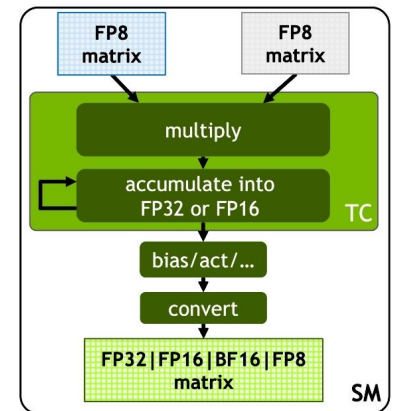
NVIDIA H100 Core



48 TFLOPS Single Precision*
24 TFLOPS Double Precision*
800 TFLOPS (FP16, Tensor Cores)*

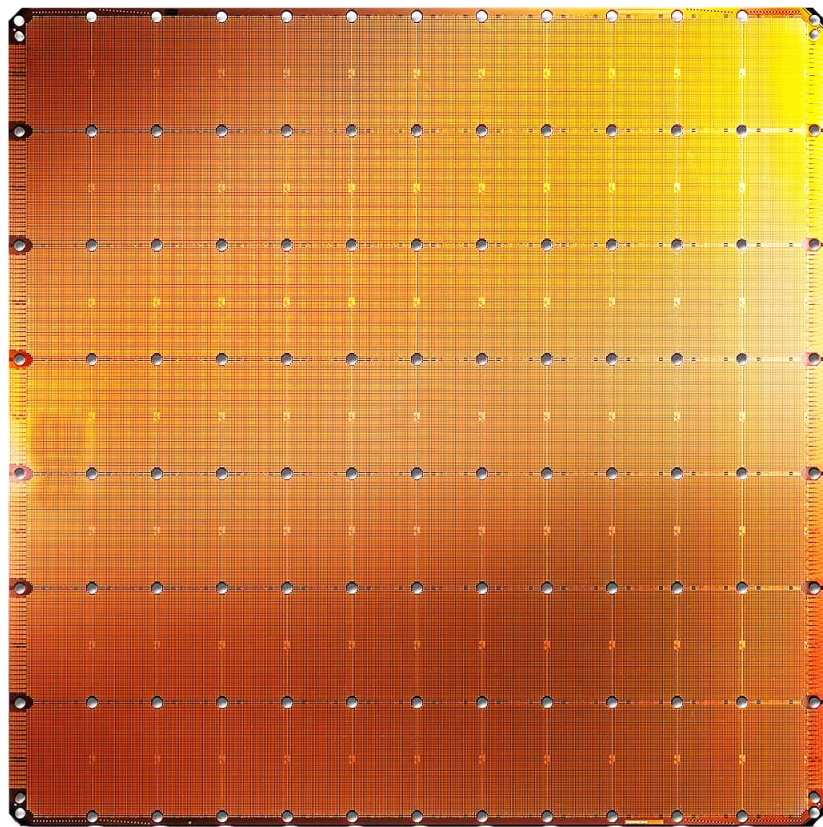


Allocate 1 bit to either range or precision



Support for multiple accumulator and output types

Cerebras's Wafer Scale Engine (2019)



Cerebras WSE

1.2 Trillion transistors

46,225 mm²

- The largest ML accelerator chip (2019)
- 400,000 cores



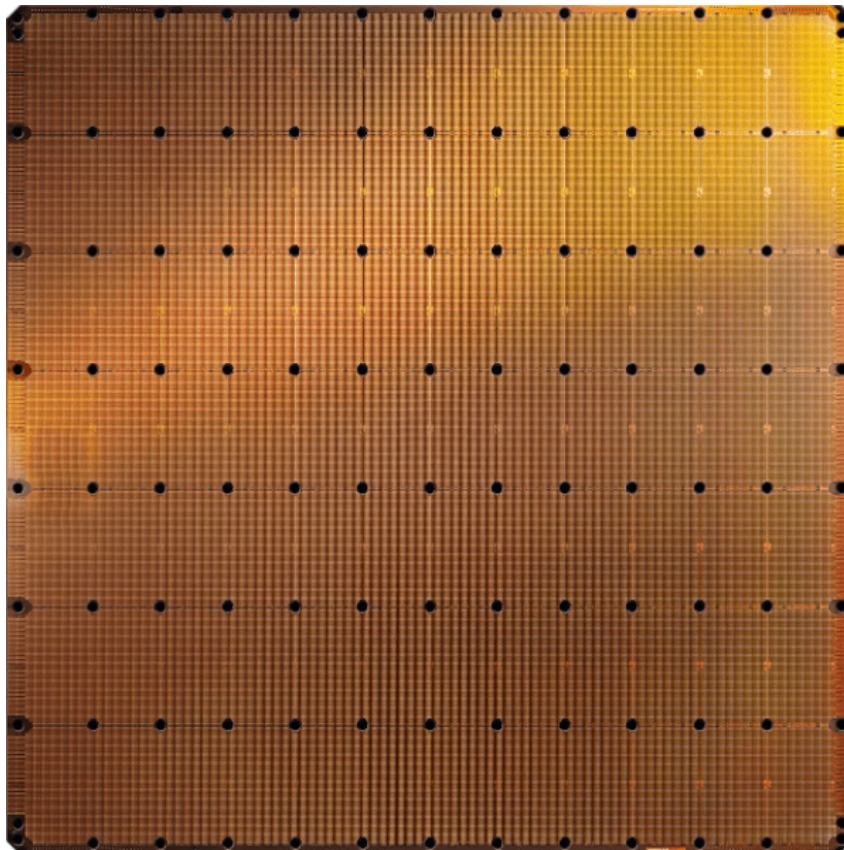
Largest GPU

21.1 Billion transistors

815 mm²

NVIDIA TITAN V

Cerebras's Wafer Scale Engine-2 (2021)



Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

- The largest ML accelerator chip (2021)
- 850,000 cores



Largest GPU
54.2 Billion transistors
826 mm²
NVIDIA Ampere GA100

SAFARI Live Seminar: Sean Lie



SAFARI Live Seminar: Sean Lie, 28 Feb 2022

Posted on January 19, 2022 by ewent

Join us for our [SAFARI Live Seminar](#) with [Sean Lie, Cerebras Systems](#)

Monday, February 28 2022 at 6:00 pm Zurich time (CET)

[Sean Lie](#), co-founder and Chief Hardware Architect at [Cerebras Systems](#)

Thinking Outside the Die: Architecting the ML Accelerator of the Future

Livestream on YouTube [Link](#)

<https://safari.ethz.ch/safari-live-seminar-sean-lie-28-feb-2022/>

Google TPU Generation I (~2016)

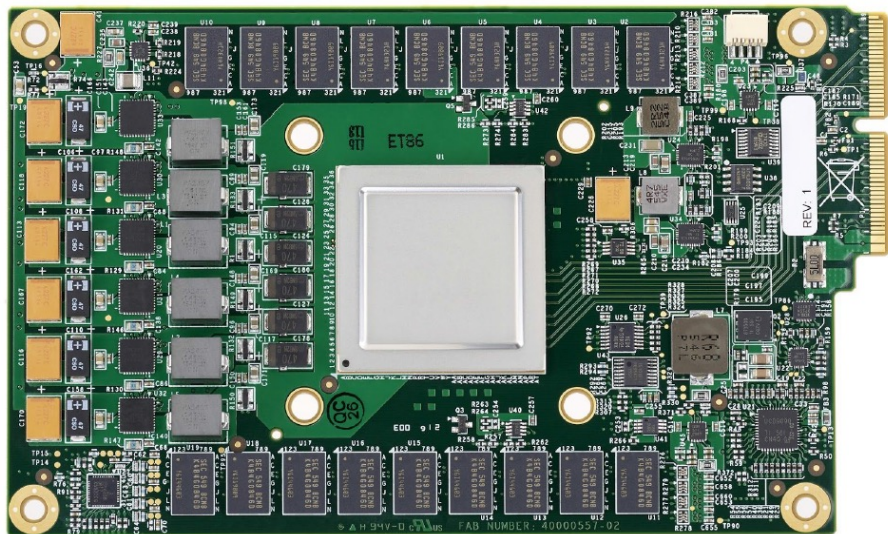


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

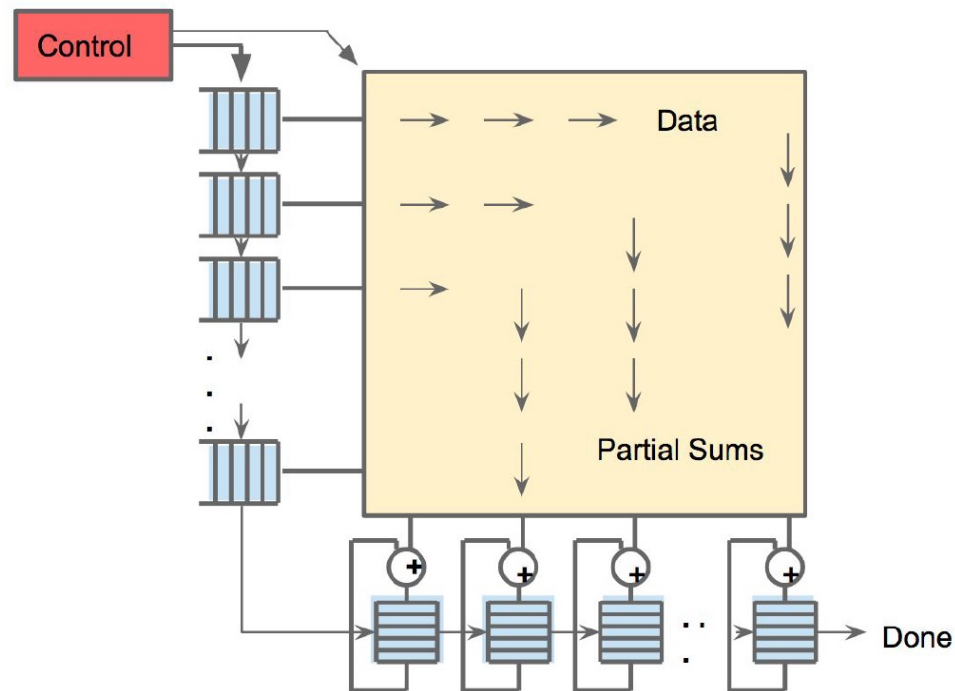
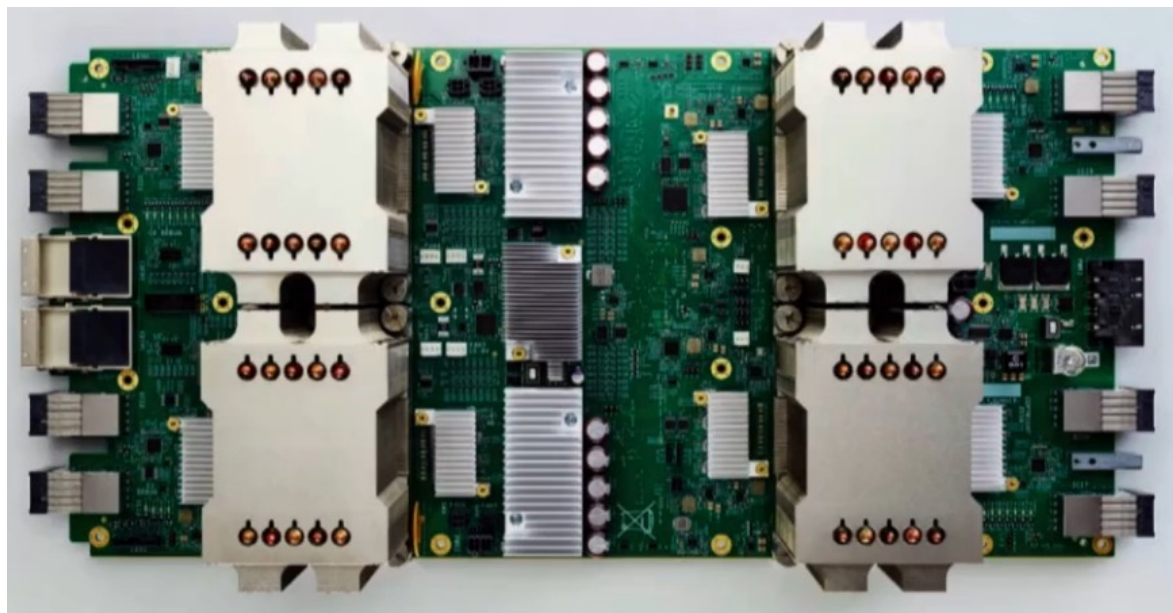


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., “In-Datacenter Performance Analysis of a Tensor Processing Unit”, ISCA 2017.

Google TPU Generation II (2017)



<https://www.nextplatform.com/2017/05/17/first-depth-look-googles-new-second-generation-tpu/>

4 TPU chips
vs 1 chip in TPU1

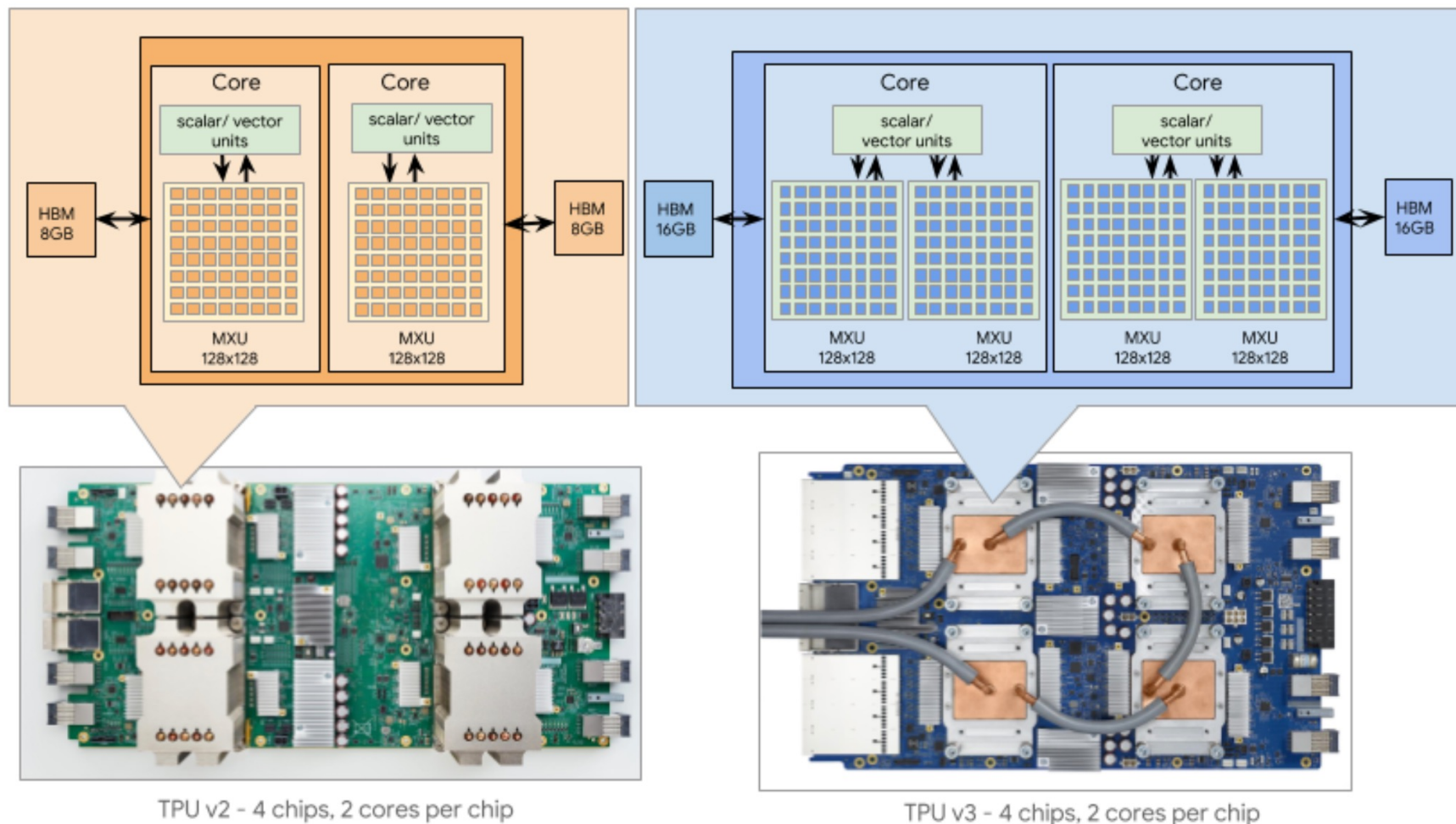
High Bandwidth Memory
vs DDR3

Floating point operations
vs FP16

45 TFLOPS per chip
vs 23 TOPS

Designed for training
and inference
vs only inference

Google TPU Generation III (2019)

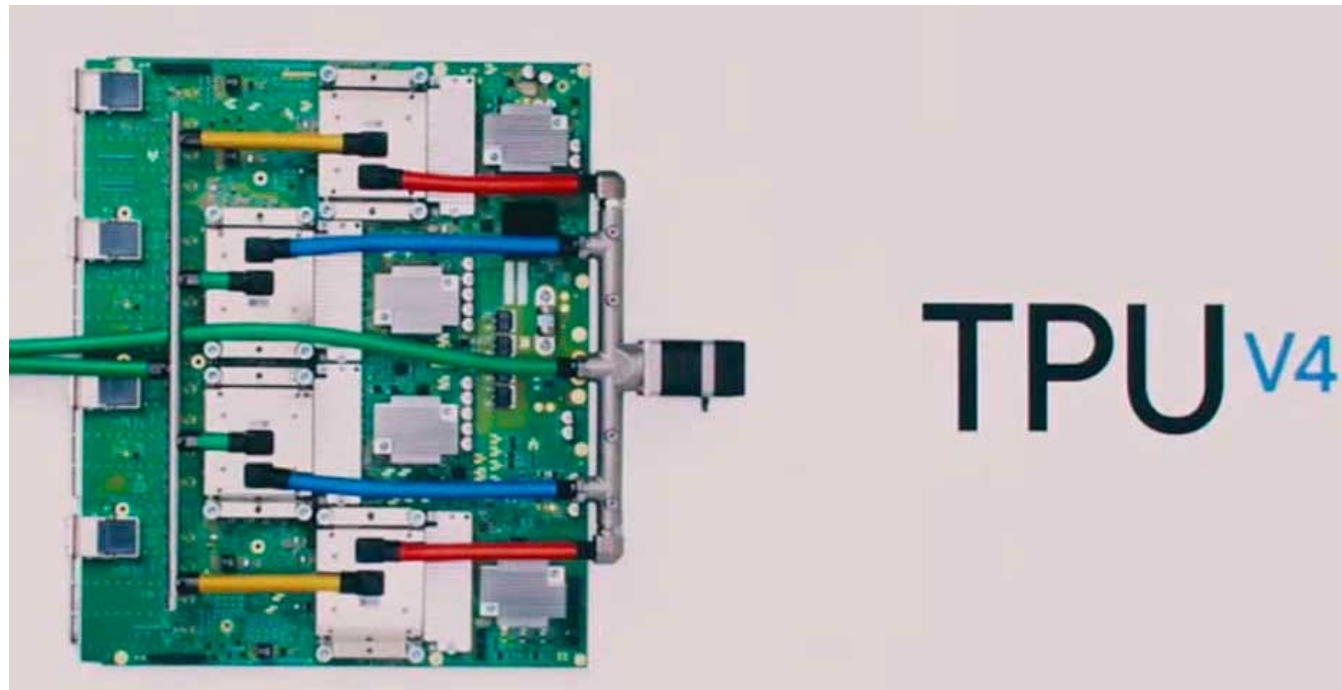


32GB HBM per chip
vs 16GB HBM in TPU2

4 Matrix Units per chip
vs 2 Matrix Units in TPU2

90 TFLOPS per chip
vs 45 TFLOPS in TPU2

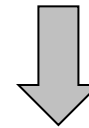
Google TPU Generation IV (2019)



New ML applications (vs. TPU3):

- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

250 TFLOPS per chip in 2021
vs 90 TFLOPS in TPU3

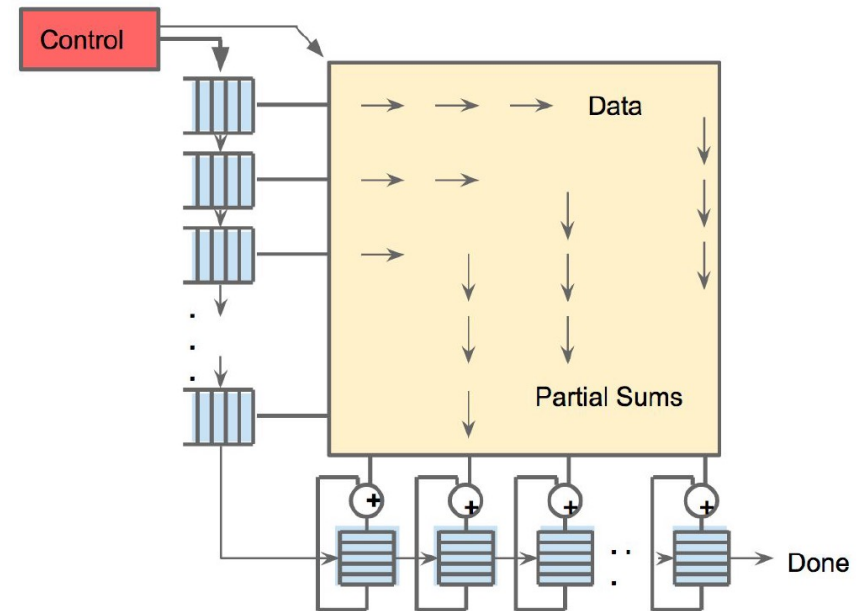


1 ExaFLOPS per board

<https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests>

An Example Modern Systolic Array: TPU (II)

As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the Unified Buffer [Kun80][Ram91][Ovt15b]. Figure 4 shows that data flows in from the left, and the weights are loaded from the top. A given 256-element multiply-accumulate operation moves through the matrix as a diagonal wavefront. The weights are preloaded, and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give the illusion that the 256 inputs are read at once, and that they instantly update one location of each of 256 accumulators. From a correctness perspective, software is unaware of the systolic nature of the matrix unit, but for performance, it does worry about the latency of the unit.



Jouppi et al., “In-Datacenter Performance Analysis of a Tensor Processing Unit”, ISCA 2017.

An Example Modern Systolic Array: TPU (III)

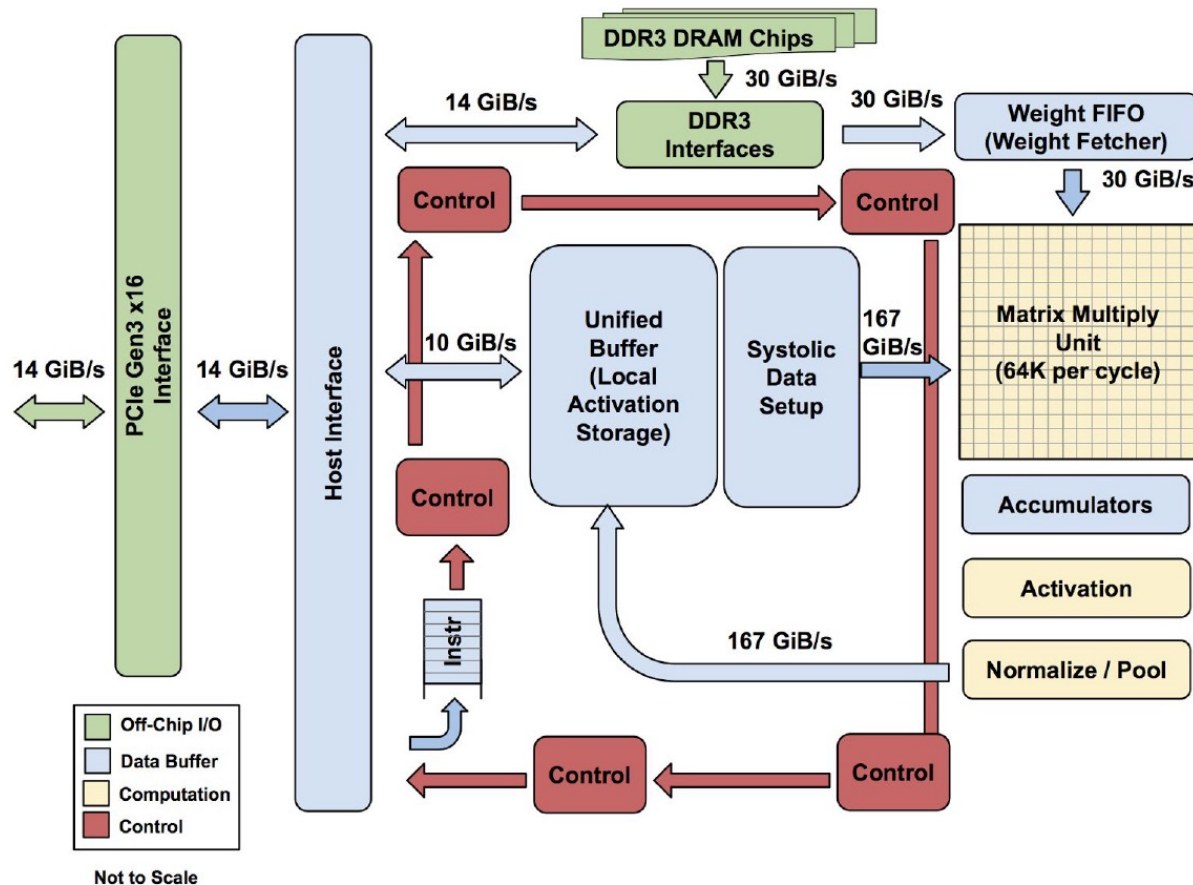
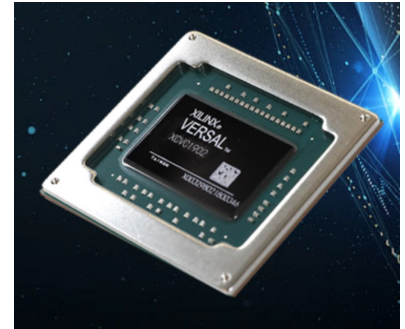


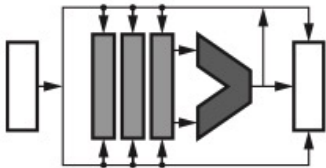
Figure 1. TPU Block Diagram. The main computation part is the yellow Matrix Multiply unit in the upper right hand corner. Its inputs are the blue Weight FIFO and the blue Unified Buffer (UB) and its output is the blue Accumulators (Acc). The yellow Activation Unit performs the nonlinear functions on the Acc, which go to the UB.

Xilinx Versal ACAP (2020) (I)

- **Three compute engines** inside the same chip
 - Different workloads, different devices

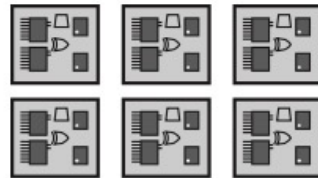


Scalar Processing



Complex Algorithms
and Decision Making

Adaptable Hardware

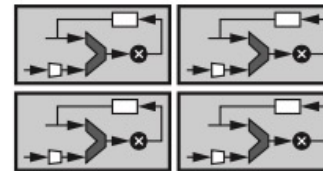


Processing of
Irregular Data Structures
Genomic Sequencing

Latency
Critical Workloads
Real-Time Control

Sensor Fusion
Pre-processing, Programmable I/O

Vector Processing (e.g., GPU, DSP)



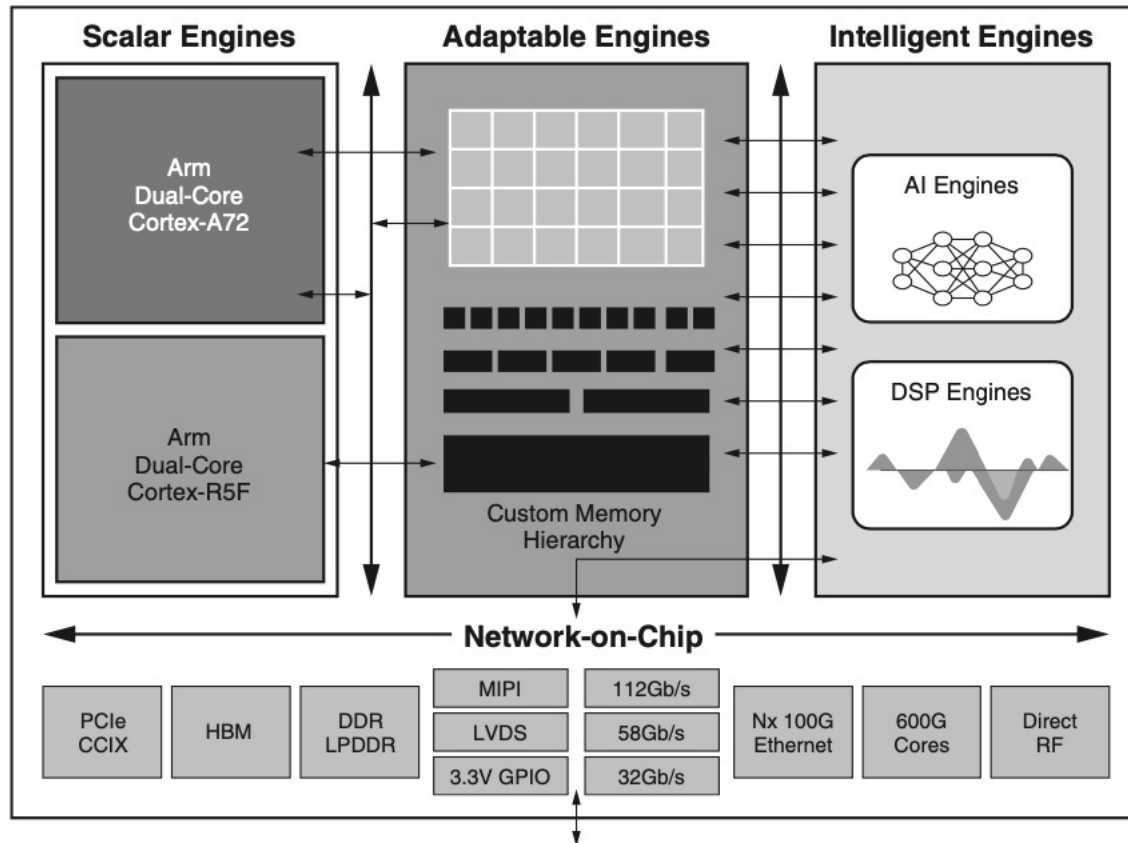
Domain-specific
Parallelism

Signal Processing
Complex Math, Convolutions

Video and
Image Processing

Xilinx Versal ACAP (2020) (II)

- **Three compute engines** inside the same chip
 - Scalar cores, reconfigurable engines, vector processors

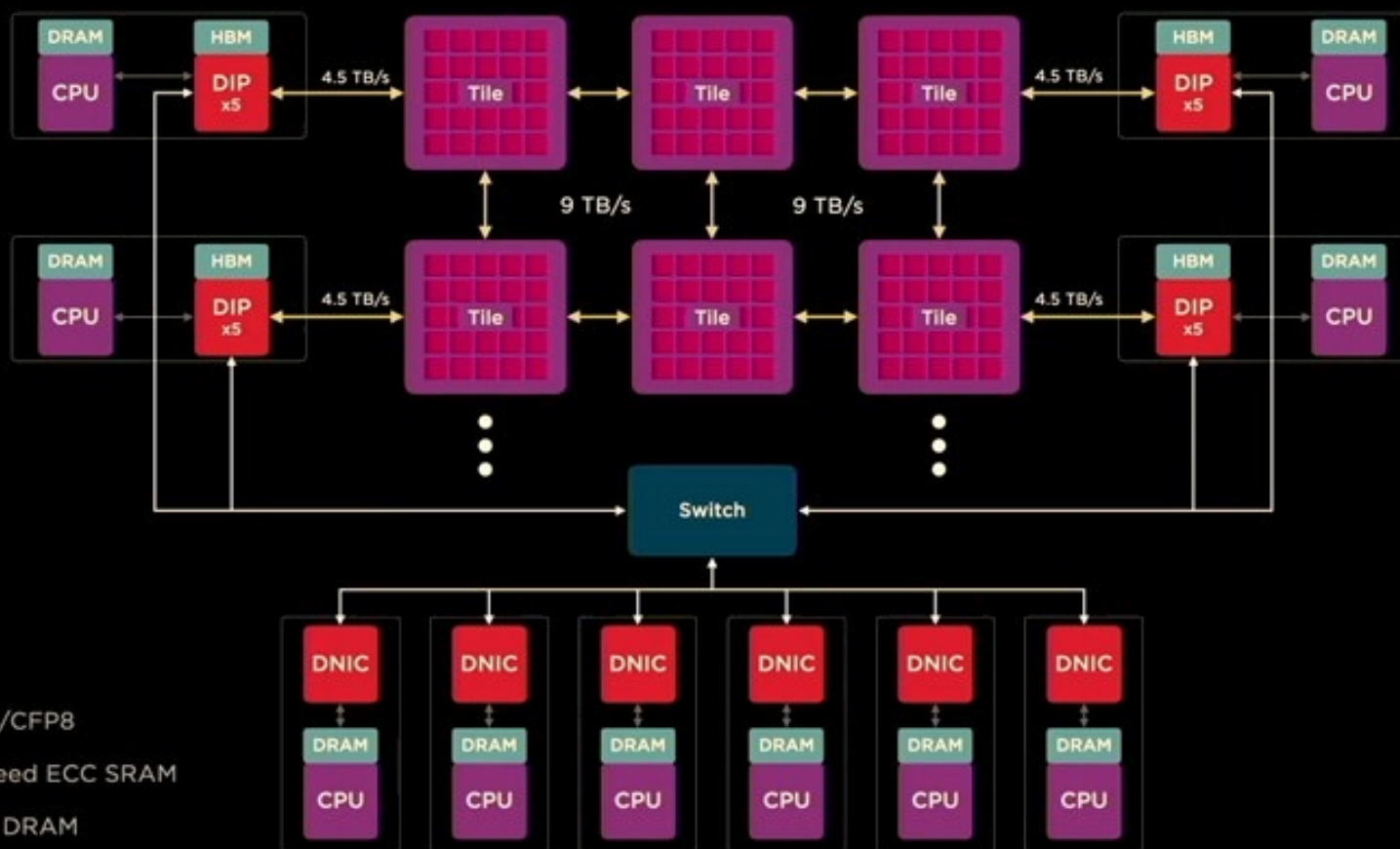


Tesla Dojo (2022) (I)



■ Tesla Dojo Chip & System

V1 Dojo Training Matrix



1 EFLOP BF16/CFP8
1.3 TB High-Speed ECC SRAM
13 TB High-BW DRAM

Tesla Dojo (2022) (II)



■ Tesla Dojo Chip & System

V1 Dojo Interface Processor

32GB High-Bandwidth Memory

- 800 GB/s Total Memory Bandwidth

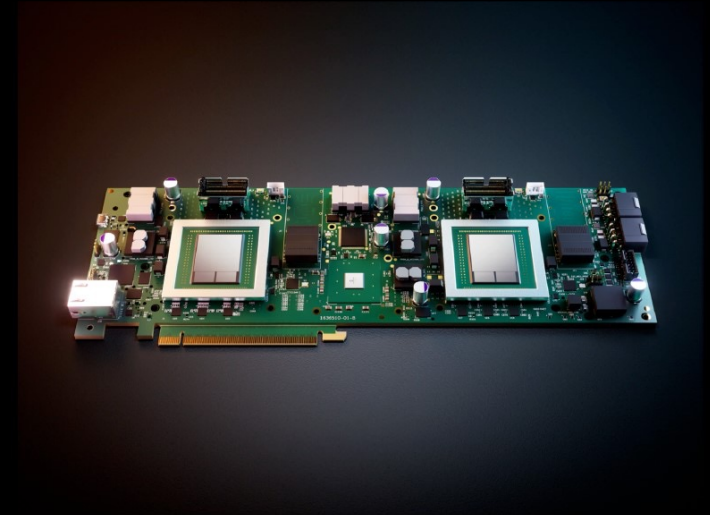
900 GB/s TTP Interface

- Tesla Transport Protocol (TTP) - Full custom protocol
- Provides full DRAM bandwidth to Training Tile

50 GB/s TTP over Ethernet (TTPoE)

- Enables extending communication over standard Ethernet
- Native hardware support

32 GB/s Gen4 PCIe Interface



Tesla Dojo (2022) (III)



■ Tesla Dojo Chip & System

D1 Chip

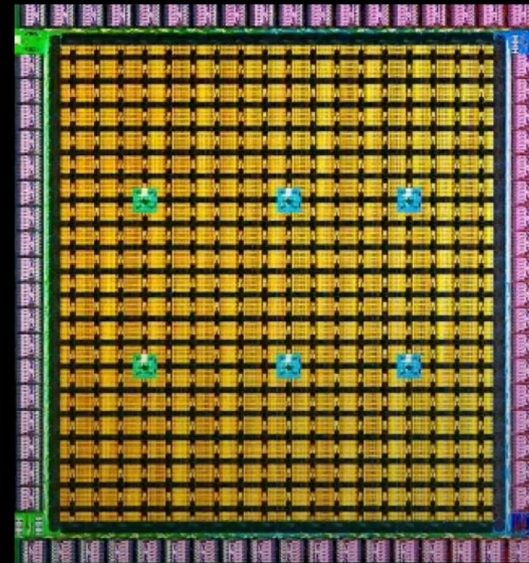
362 TFLOPs BF16/CFP8

22.6 TFLOPs FP32

10TBps/dir. On-Chip Bandwidth

4TBps/edge. Off-Chip Bandwidth

400W TDP



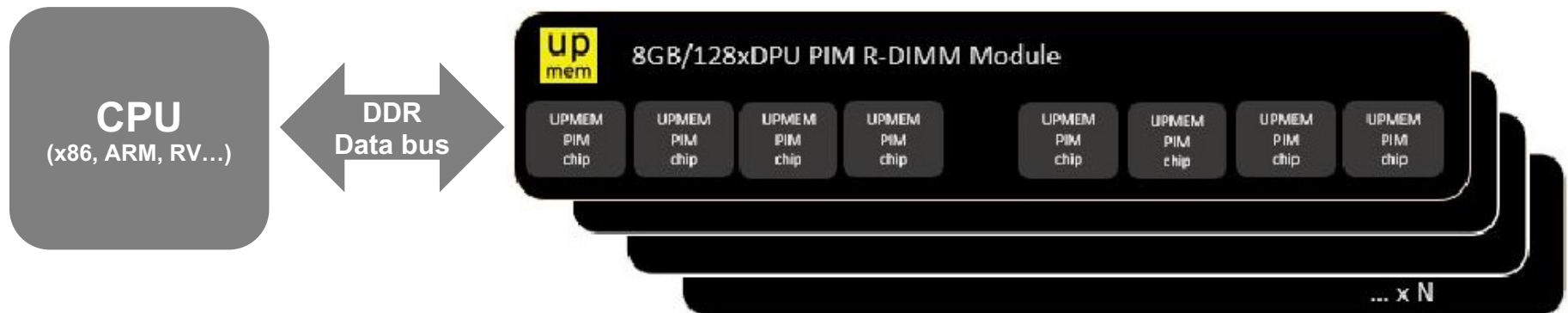
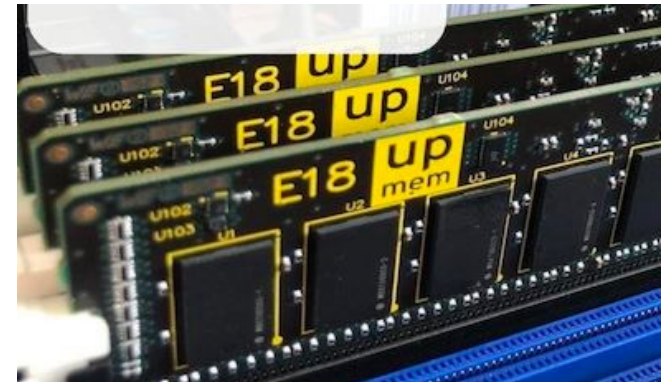
645mm²
7nm Technology

50 Billion
Transistors

11+ Miles
Of Wires

UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
 - DDR4 R-DIMM modules
 - 8GB+128 DPUs (16 PIM chips)
 - Standard 2x-nm DRAM process
 - **Large amounts of** compute & memory bandwidth



Samsung Function-in-Memory DRAM (2021)



Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021

Audio



Share



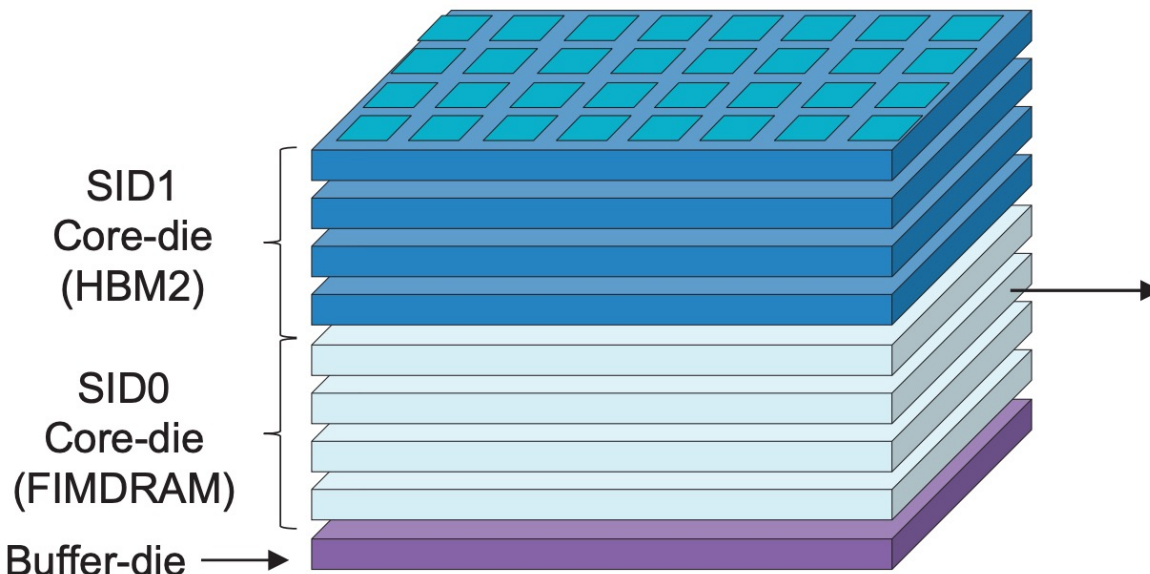
The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%

Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power – the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, "Our groundbreaking HBM-PIM is the industry's first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications."

Samsung Function-in-Memory DRAM (2021)

■ FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

Chip Specification

128DQ / 8CH / 16 banks / BL4

32 PCU blocks (1 FIM block/2 banks)

1.2 TFLOPS (4H)

**FP16 ADD /
Multiply (MUL) /
Multiply-Accumulate (MAC) /
Multiply-and- Add (MAD)**

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang², Shinhaeng Kang¹, Yuhwan Ro³, Seungwoo Seo³, JoonHo Song³, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwaseong, Korea

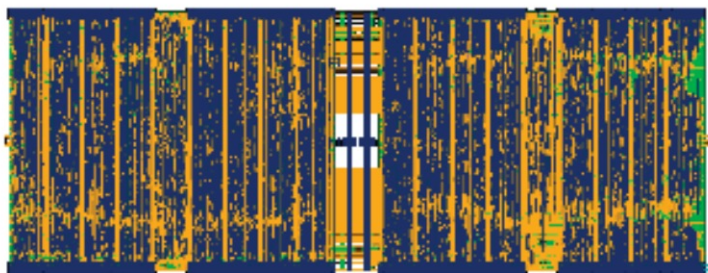
²Samsung Electronics, San Jose, CA

³Samsung Electronics, Suwon, Korea

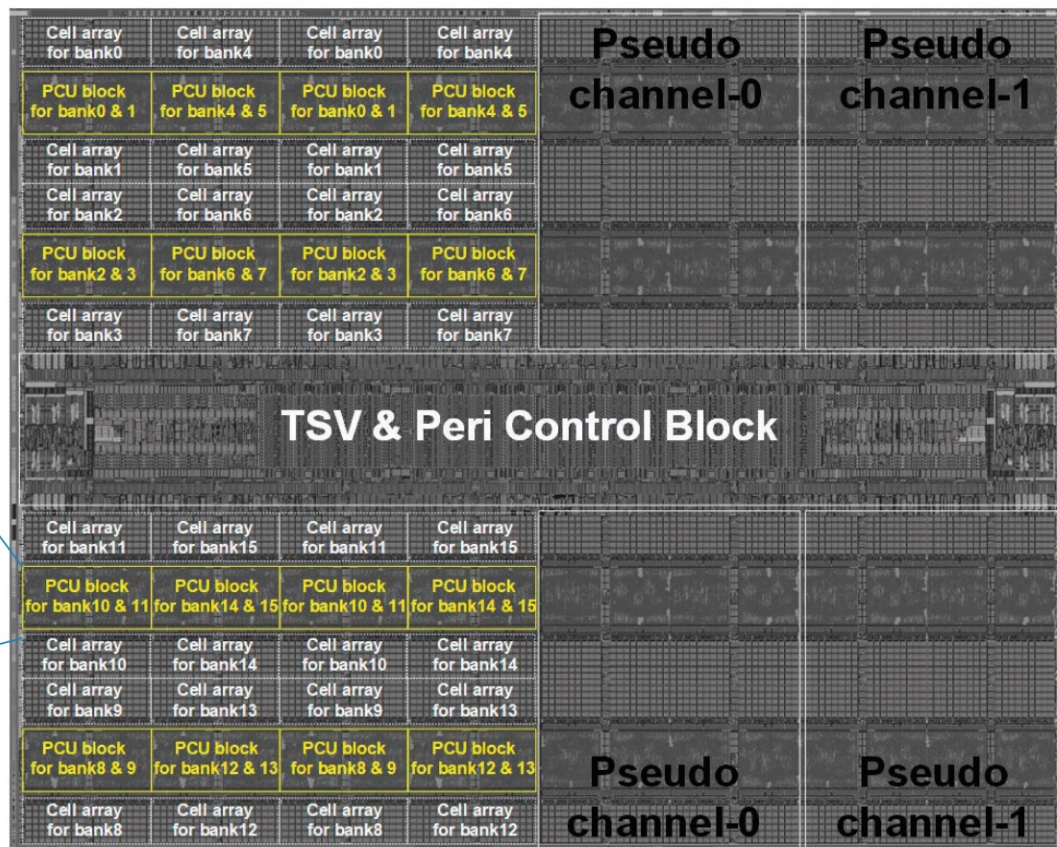
Samsung Function-in-Memory DRAM (2021)

Chip Implementation

- Mixed design methodology to implement FIMDRAM
 - Full-custom + Digital RTL



[Digital RTL design for PCU block]



ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyeon Choi¹, Hyun-Sung Shim¹, Jin Kim¹, BengSeng Phuah¹, HyounMin Kim¹, Myeong Jun Song¹, Ahn Chai¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang², Shintae Kang³, Yulwan Ro³, Seungwoo Seo³, JoonHo Song³, Jaeyoun Yoon¹, Kyomin Sohn¹, Nam Sung Kim¹

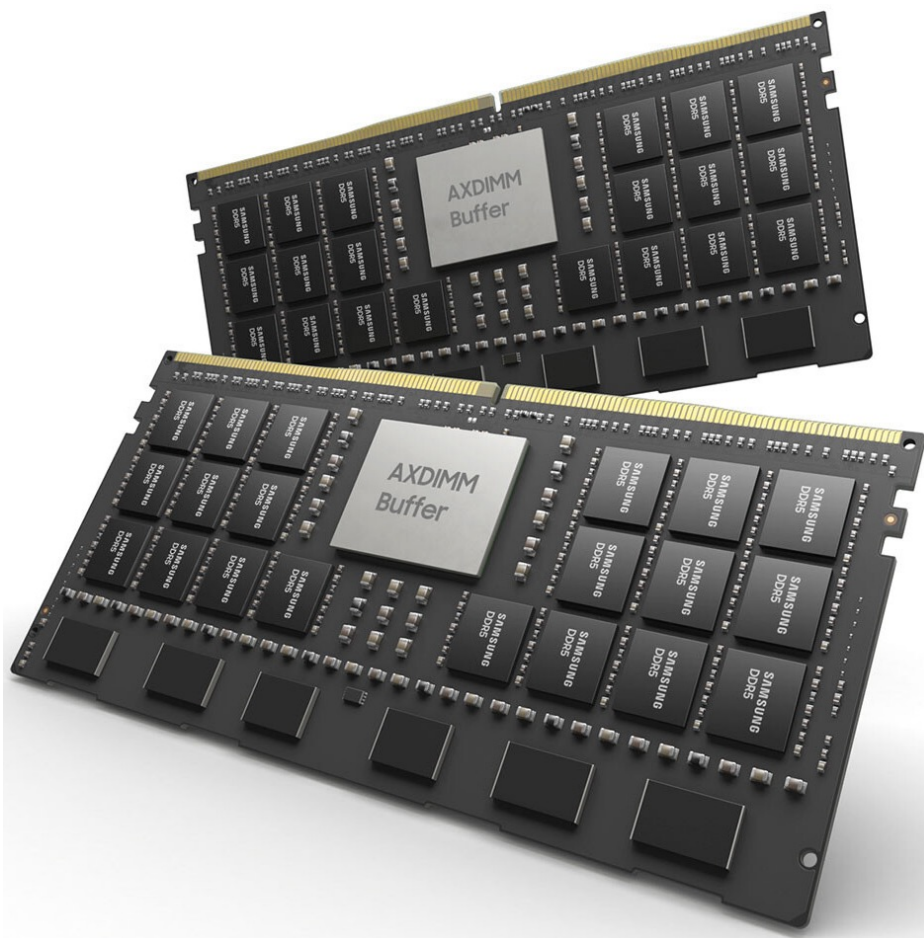
¹Samsung Electronics, Hwaseong, Korea

²Samsung Electronics, San Jose, CA

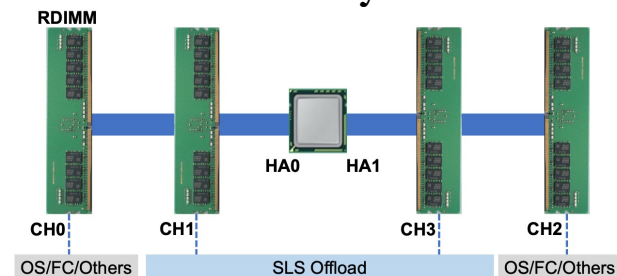
³Samsung Electronics, Suwon, Korea

Samsung AxDIMM (2021)

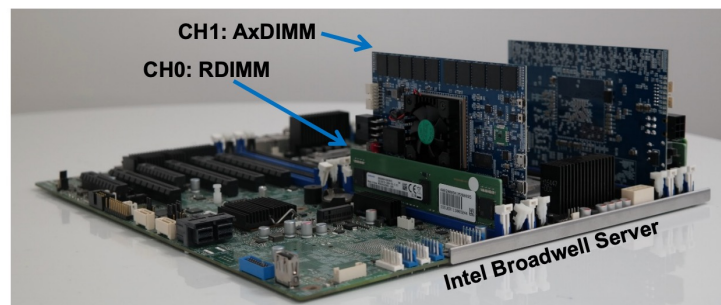
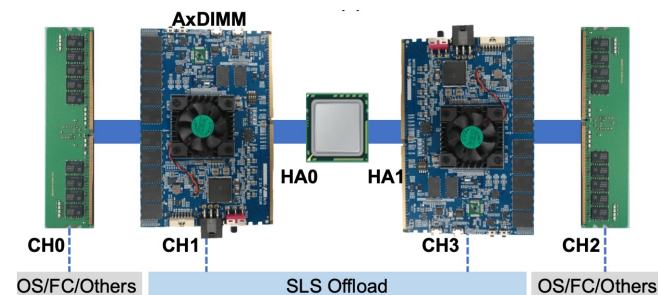
- DIMM-based PIM
 - DLRM recommendation system



Baseline System



AxDIMM System



SK Hynix Accelerator-in-Memory (2022)

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022



Seoul, February 16, 2022

SK hynix (or “the Company”, www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

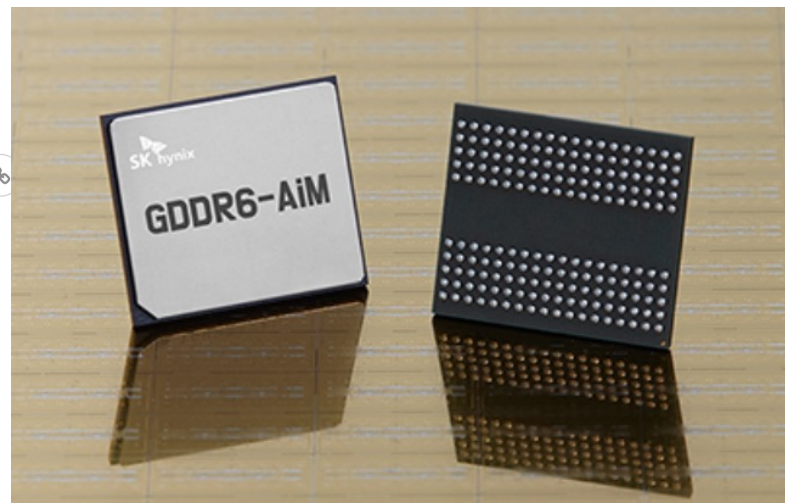
**PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory*

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world’s most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

**ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of “Intelligent Silicon for a Sustainable World”*

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.



11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

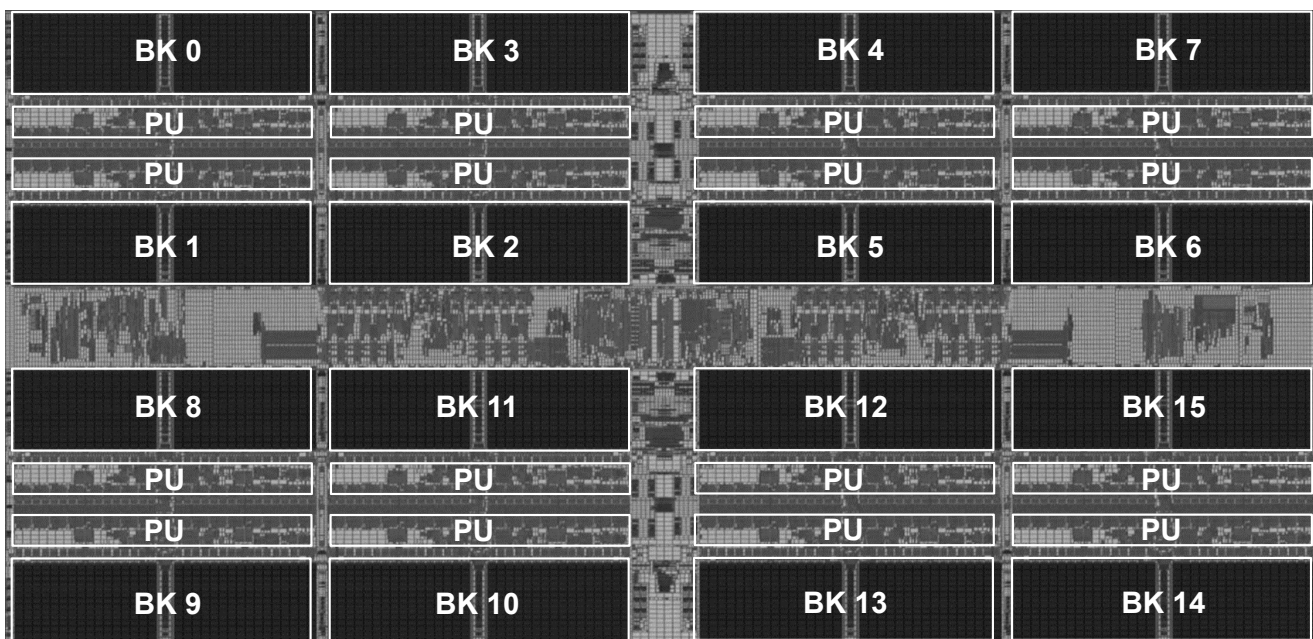
Seongju Lee, SK hynix, Icheon, Korea

In Paper 11.1, SK Hynix describes a 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

SK Hynix AiM: Chip Implementation (2022)

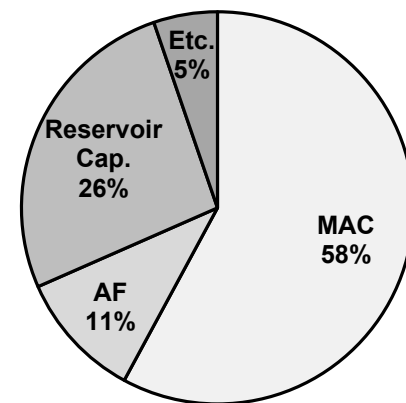
- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph



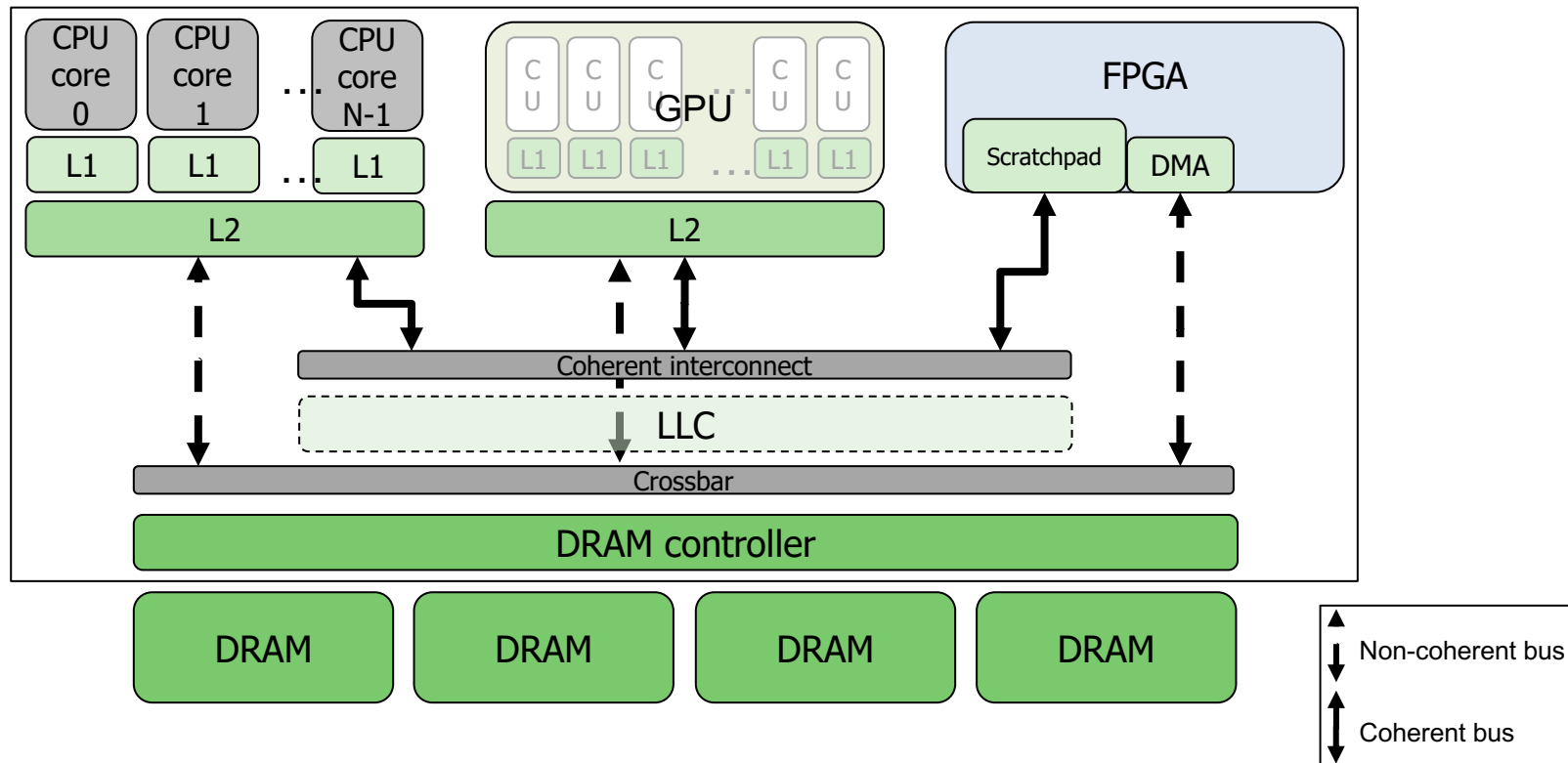
1 Process Unit (PU) Area

Total	0.19mm ²
MAC	0.11mm ²
Activation Function (AF)	0.02mm ²
Reservoir Cap.	0.05mm ²
Etc.	0.01mm ²

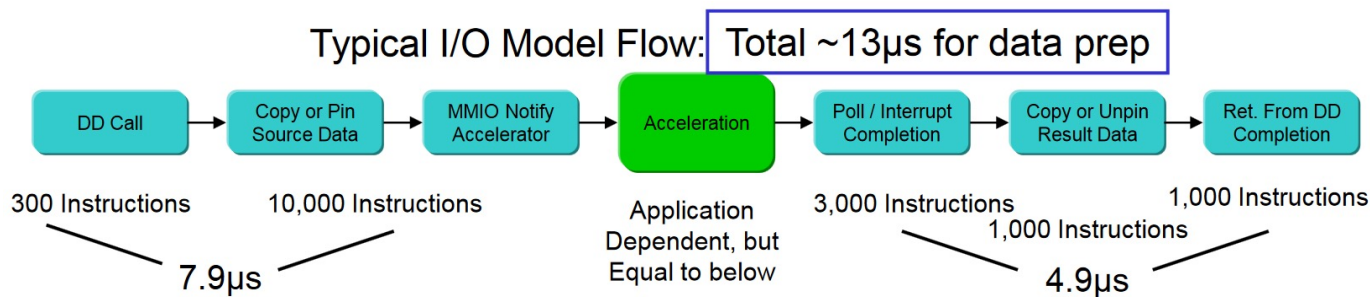
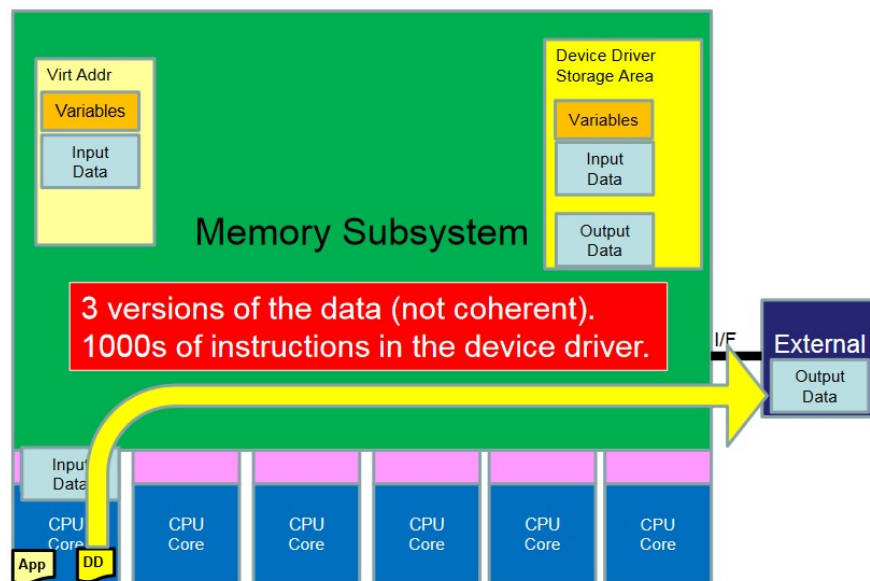


Heterogeneous Computing Systems

- The end of Moore's law created **the need for heterogeneous systems**
 - More **suitable devices** for each type of workload
 - Increased **performance and energy efficiency**



Background: Traditional I/O Technology

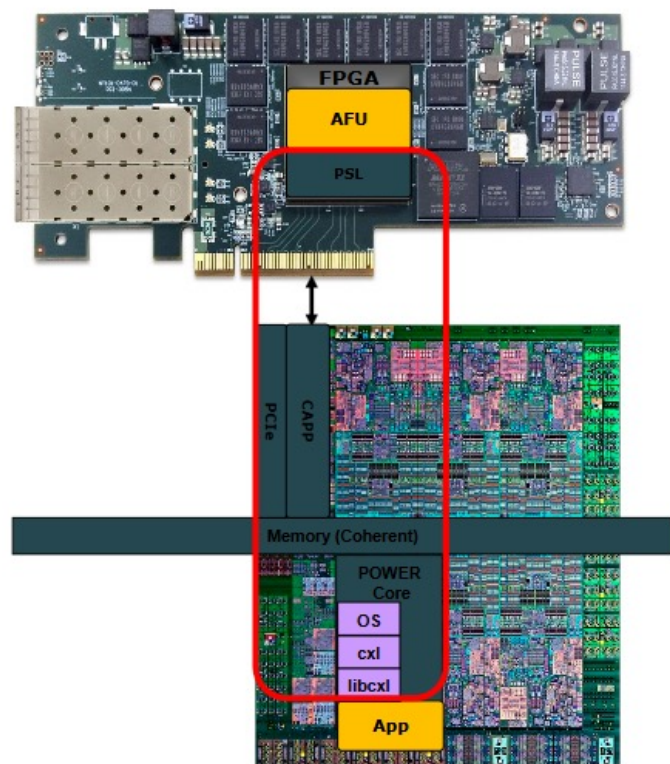


Dionysios Diamantopoulos, IBM Research – Zurich, COOL Chips 2018

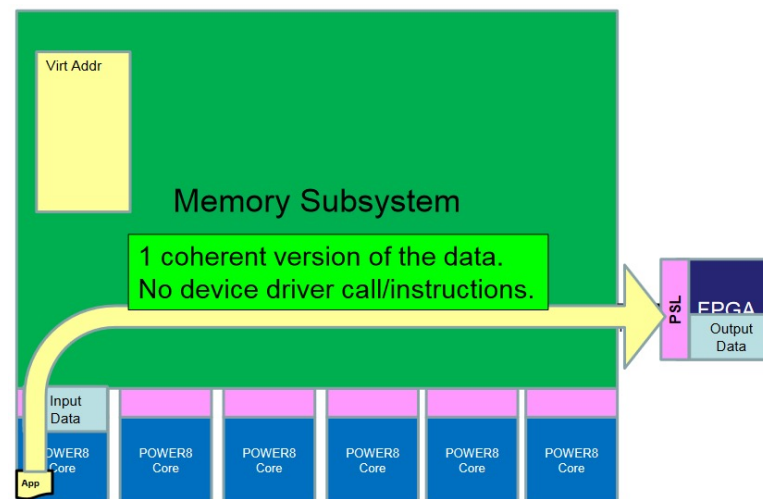


CAPI/OpenCAPI Overview

- CAPI/CAPI2 (Coherent Accelerator Processor Interface)
- OpenCAPI

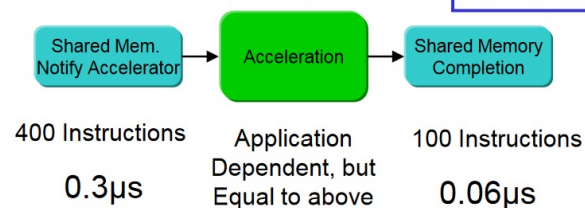


POWER8 - POWER9 Processor



Flow with a CAPI Model:

Total 0.36 μ s



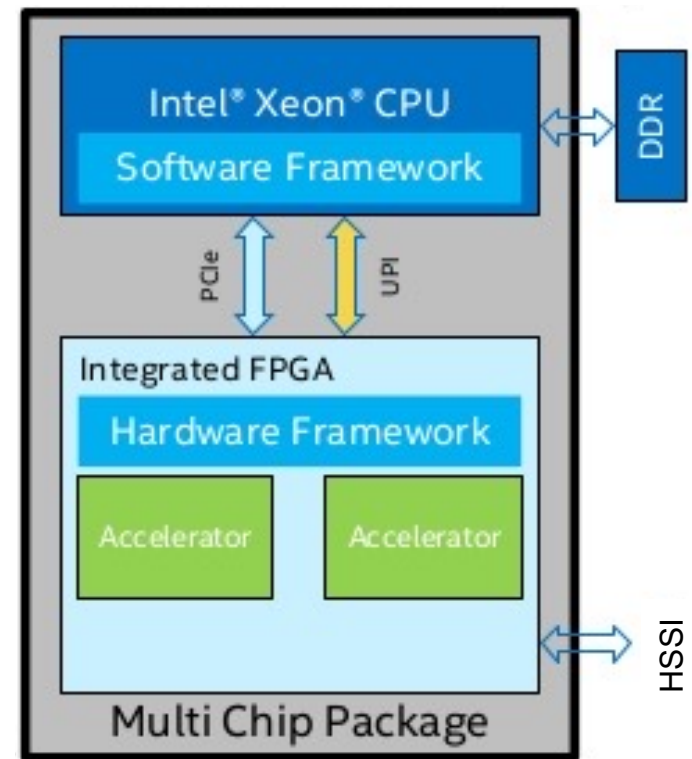
Dionysios Diamantopoulos, IBM Research – Zurich, COOL Chips 2018



Collaborative Computing on CPU+FPGA

- Traditionally, accelerators (GPUs, FPGAs, etc.) have been used as *offload* engines
- Heterogeneous architectures moving towards tighter integration
 - ❑ Unified memory
 - ❑ System-wide atomics
- Tighter integration allows fine-grained collaboration

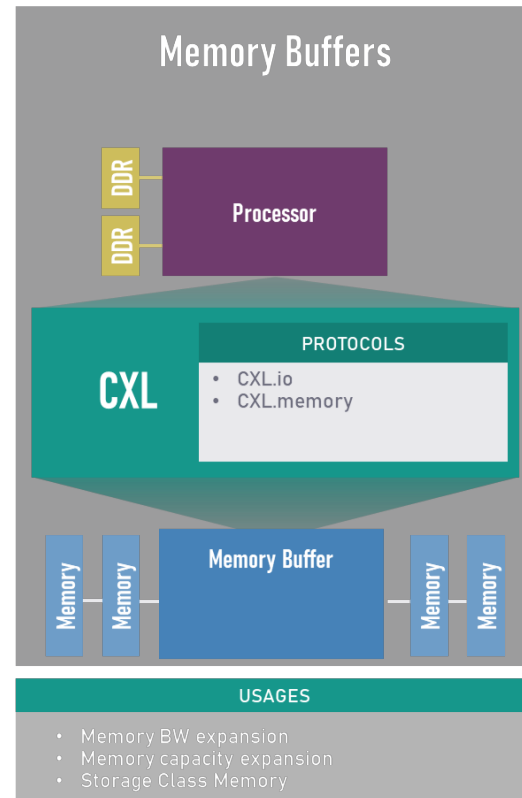
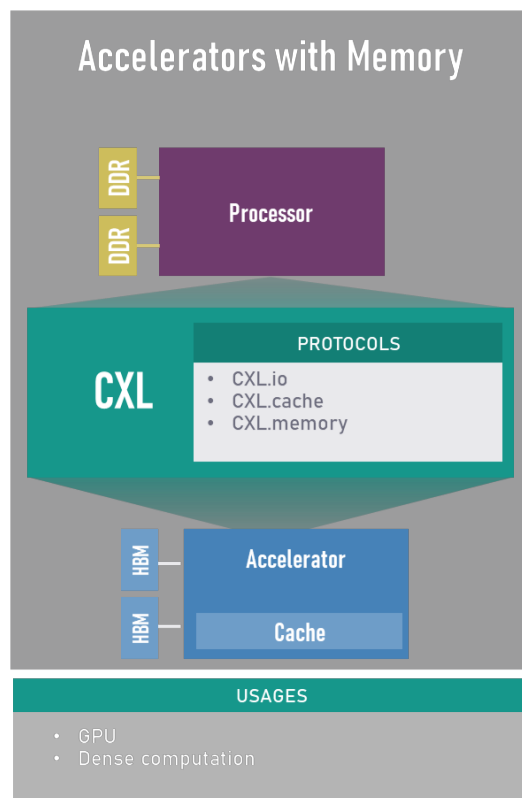
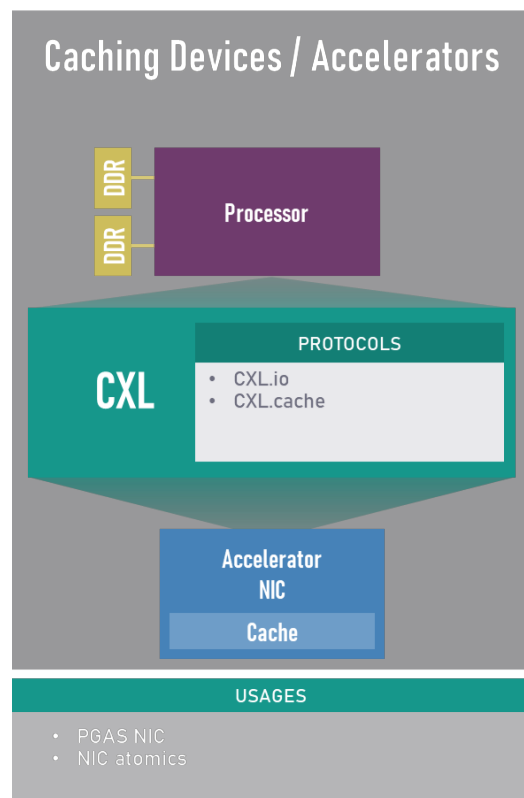
Key challenge: identify the best CPU-FPGA collaboration strategy



Intel Xeon + FPGA Integrated Platform (MCP)

Compute Express Link (CXL)

- Compute Express Link (CXL) is an open industry standard interconnect offering **high-bandwidth, low-latency connectivity between host processor and devices** such as accelerators, memory buffers, and smart I/O devices



Key Takeaways

- This P&S is aimed at improving your
 - **Knowledge** in Computer Architecture and Heterogeneous Systems
 - **Technical skills** in programming heterogeneous architectures
 - **Critical thinking and analysis**
 - **Interaction** with a nice group of researchers
 - Familiarity with key **research directions**
 - **Technical presentation** of your project

Key Goal

(Learn how to) take advantage of
existing heterogeneous devices
by programming them,
analyzing workloads, proposing
offloading/scheduling techniques...

Prerequisites of the Course

- Digital Design and Computer Architecture (or equivalent course)
 - <https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule>
 - <https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule>
- Familiarity with C/C++ programming
 - FPGA implementation or GPU programming (desirable)
- Interest in
 - computer architectures and computing paradigms
 - discovering why things do or do not work and solving problems
 - making systems efficient and usable

Course Info: Who Are We? (I)



■ Onur Mutlu

- ❑ Full Professor @ ETH Zurich ITET (INFK), since September 2015
- ❑ Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- ❑ PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- ❑ <https://people.inf.ethz.ch/omutlu/>
- ❑ omutlu@gmail.com (Best way to reach me)
- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>

■ Research and Teaching in:

- ❑ Computer architecture, computer systems, hardware security, bioinformatics
- ❑ Memory and storage systems
- ❑ Hardware security, safety, predictability
- ❑ Fault tolerance
- ❑ Hardware/software cooperation
- ❑ Architectures for bioinformatics, health, medicine
- ❑ ...

Course Info: Who Are We? (II)

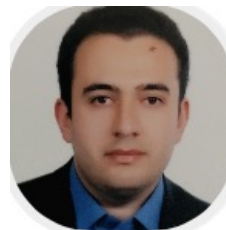
- Lead Supervisor:

- Dr. Juan Gómez Luna



- Supervisors:

- Dr. Mohammed Alser
- Dr. Behzad Salami
- Dr. Mohammad Sadr
- Joel Lindegger



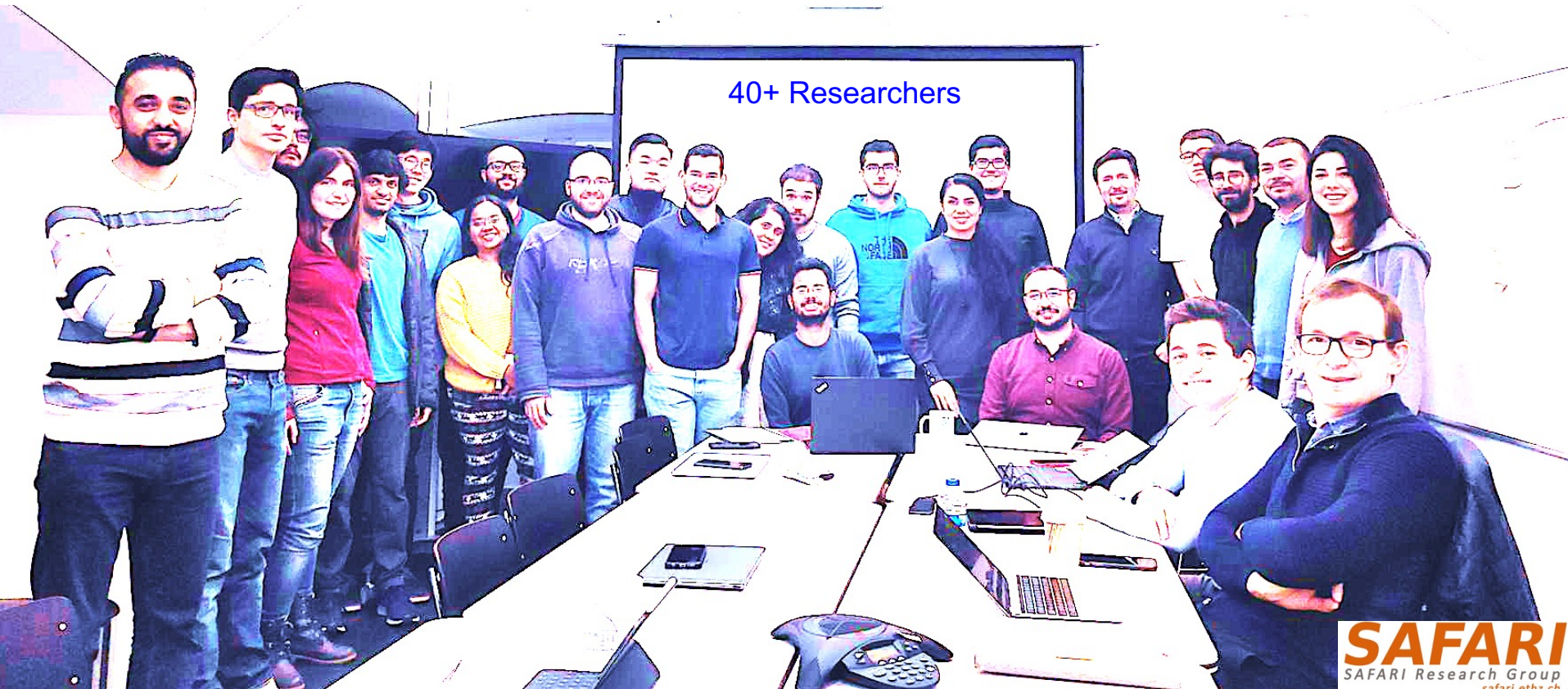
- Get to know us and our research

- <https://safari.ethz.ch/safari-group/>

Onur Mutlu's SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

<https://safari.ethz.ch/safari-newsletter-january-2021/>



SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

<https://safari.ethz.ch>

SAFARI Newsletter December 2021 Edition

- <https://safari.ethz.ch/safari-newsletter-december-2021/>

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich



View in your browser
December 2021



SAFARI Live Seminars (I)


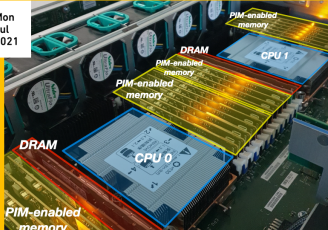
SAFARI Live Seminars in Computer Architecture

Dr. Juan Gómez Luna, ETH Zurich

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

SAFARI
SAFARI Research Group

12 Mon Jul 2021


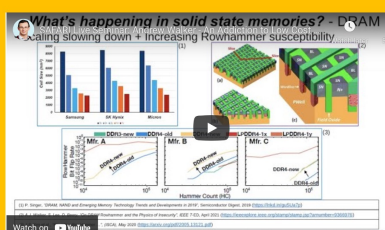
SAFARI Live Seminars in Computer Architecture

Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems

An Addition to Low Cost Per Memory Bit – How to Recognize It and What to Do About It

SAFARI
SAFARI Research Group

19 Mo Jul 2021


SAFARI Live Seminars in Computer Architecture

Geraldo F. Oliveira, ETH Zurich

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

SAFARI
SAFARI Research Group

22 Do Jul 2021



Near-Data Processing (2/2)

UPMEM (2019) Samsung HBM-PIM (2021)

Near-DRAM-banks processing for general-purpose computing

Near-DRAM-banks processing for neural networks

0.9 TOPS compute throughput¹ 1.2 TFLOPS compute throughput²

The goal of Near-Data Processing (NDP) is to mitigate data movement

SAFARI © 2021 ETH Zurich, The Data Processing Research Institute, DSI, 2021. All rights reserved. This work is licensed under a Creative Commons Attribution 4.0 International License. For more information, see the License at <http://creativecommons.org/licenses/by/4.0/>.


SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Efficient DNN Training at Scale: from Algorithms to Hardware

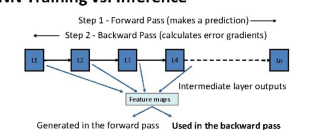
SAFARI
SAFARI Research Group

5 Do Aug 2021



DNN Training vs. Inference

Step 1 - Forward Pass (makes a prediction)
Step 2 - Backward Pass (calculates error gradients)



Generated in the forward pass Used in the backward pass

DNN training requires stashing feature maps for the backward pass (not required in inference)


SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Power Management Mechanisms in Modern Microprocessors and Their Security Implications

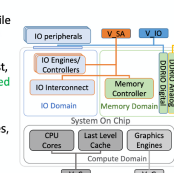
SAFARI
SAFARI Research Group

16 Mo Aug 2021



Overview of a Modern SoC Architecture

- 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
- Several voltage sources exist, and some of them are shared between domains
- IO controllers and engines, IO interconnect, memory controller, and DDRIO typically each has an independent clock




SAFARI Live Seminars in Computer Architecture

Ataberk Olgun, TOBB & ETH Zurich

QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips

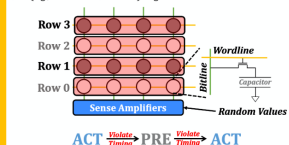
SAFARI
SAFARI Research Group

15 Mi Sep 2021



Using QUAC to Generate Random Values

Use QUAC to activate DRAM rows that are initialized with conflicting data (e.g., two '1's and two '0's) to generate random values



Row 3
Row 2
Row 1
Row 0

Sense Amplifiers

ACT

SAFARI © kasirga


SAFARI Live Seminars in Computer Architecture

Minesh Patel, ETH Zurich

Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

SAFARI
SAFARI Research Group

21 Tues Sep 2021



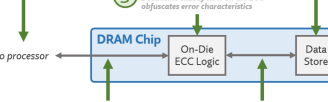
Position Paper (Ongoing) Arguing for increased transparency of DRAM reliability characteristics

REAPER (ISCA'17) Understand the basic properties of DRAM data-retention errors

BEER (MICRO'20, best paper) Determine exactly how on-die ECCs adjust error characteristics

HARP (MICRO'21) Understand how errors appear and how to identify at-risk bits

EIN (DSN'19, best paper) Understand and recover the error characteristics beneath on-die ECC




SAFARI Live Seminars in Computer Architecture

Christina Giannoula, National Technical University of Athens

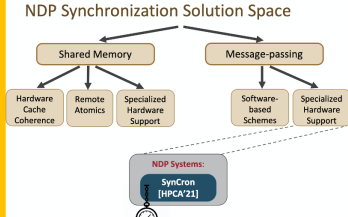
Efficient Synchronization Support for Near-Data-Processing Architectures

SAFARI
SAFARI Research Group

27 Mo Sep 2021



NDP Synchronization Solution Space




SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Security Implications of Power Management Mechanisms in Modern Processors, Current Studies and Future Trends

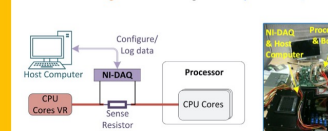
SAFARI
SAFARI Research Group

4 Mo Okt 2021



Experimental Methodology

- We experimentally study three modern Intel processors
 - Haswell, Coffee Lake, and Cannon Lake
- We measure voltage and current using a Data Acquisition card (NI-DAQ)



<https://safari.ethz.ch/safari-seminar-series/>

SAFARI Live Seminars (II)

SAFARI Live Seminars in Computer Architecture

Nastaran Hajinazar, ETH Zurich
Data-Centric and Data-Aware Frameworks for Fundamentally Efficient Data Handling in Modern Computing Systems

27 Wed Oct 2021

Overview of Our Approach

Data and the efficient computation of data should be the ultimate priority of the system

- **Data-Centric Architectures**
 - Enable computation with minimal data movement
 - Compute where data resides
- **Data-Aware Architectures**
 - Understand what they can do with and to each piece of data
 - Make use of different properties of data to improve performance, efficiency, etc.

SAFARI

SAFARI Live Seminar: Nastaran Hajinazar 27 Oct 2021
Posted on October 1, 2021 by ewent

SAFARI Live Seminars in Computer Architecture

Sergei Mangul, Mangul Lab, USC

Opportunities and challenges of computational data-driven immunology

11 Thu Nov 2021

Opportunities and challenges of computational data-driven immunology

Sergei Mangul, Ph.D
Assistant Professor,
University of Southern California

https://mangul-lab.usc.edu/athub/id/

ETH zürich

SAFARI Live Seminar: Sergei Mangul 11 Nov 2021
Posted on November 5, 2021 by ewent

SAFARI Live Seminar-CODIC: A Low-Cost Substrate for Enabling Customizable DRAM Internal Circuit Timings

• **CODIC** substrate enables greater control over DRAM internal circuit timings

• **CODIC** is an efficient and low-cost way to enable new functionalities and optimizations in DRAM

• **CODIC** controls four key signals that orchestrate DRAM internal circuit timings

- **wordline (wl)**: Connects DRAM cells to bitlines
- **sense_p** and **sense_n**: Trigger sense amplifiers
- **EQ**: Triggers the logic that prepares a DRAM bank for the next access

Watch on YouTube

14

SAFARI Live Seminar: Lois Orosa, 10 Feb 2022
Posted on January 16, 2022 by ewent

Join us for our next SAFARI Live Seminar with Lois Orosa.
Thursday, February 10 at 5:00 pm Zurich time (CET)

SAFARI Live Seminars in Computer Architecture

Damla Senol Cali, Bionano Genomics

Accelerating Genome Sequence Analysis via Efficient Hardware/Algorithm Co-Design

7 Sun Nov 2021

Our Goal & Approach

- **Our Goal:** Accelerating genome sequence analysis by efficient hardware/algorithm co-design
- **Our Approach:**
 - (1) Analyze the multiple steps and the associated tools in the genome sequence analysis pipeline,
 - (2) Expose the tradeoffs between accuracy, performance, memory usage and scalability, and
 - (3) Co-design fast and efficient algorithms along with scalable and energy-efficient customized hardware accelerators for the key bottleneck steps of the pipeline

SAFARI

SAFARI Live Seminar: Damla Senol Cali 07 Nov 2021
Posted on October 18, 2021 by ewent

SAFARI Live Seminar - Pythia: A Customizable HW Prefetching Framework Using Online Reinforcement Learning

Rahul Bera, ETH Zurich

Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Brief Overview of Pythia

Pythia formulates prefetching as a reinforcement learning problem

State (S_t) → Agent → Action (A_t) → Environment → Reward (R_{t+1})

Environment → Prefetcher → Processor & Memory Subsystem → Reward

Features of memory request to address A (e.g. PC) → Prefetch from address A+offset (Q)

Watch on YouTube

SAFARI

SAFARI Live Seminar: Rahul Bera 20 Dec 2021

SAFARI Live Seminars in Computer Architecture

Sean Lie, Cerebras

Thinking Outside the Die: Architecting the ML Accelerator of the Future

Livestream on YouTube: Feb 28, 2022 18:00 Zurich time

Thinking Outside the Die: Architecting the ML Accelerator of the Future

Sean Lie
Co-founder & Chief HW Architect, Cerebras

Posted on January 19, 2022 by ewent

Join us for our SAFARI Live Seminar with Sean Lie, Cerebras Systems
Monday, February 28 2022 at 6:00 pm Zurich time (CET)

SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Machine Learning Tools in Action

8 Mo Nov 2021

RL Scope: Cross-Stack Profiling for Deep Reinforcement Learning Workloads

Watch on YouTube

SAFARI

SAFARI Live Seminar: Gennady Pekhimenko 08 Nov 2021
Posted on November 1, 2021 by ewent

SAFARI Live Seminar - Introduction to the UPMEM DPU Architecture

UPMEM PIM DRAM (1/2)

8 x 32-bit CPU added to a 4Gb DRAM die:

- First Gen: 8 x CPU @450MHz, 8 x 64 MB banks (1 CPU for 1 bank)
- Second Gen: 8 x CPU @600MHz, 16 x 32 MB banks (1 CPU for 2 banks), secure Enclave

Multi-threaded CPU:

- In order execution at the thread level
- Out of order execution between threads when executing DMA instructions

Offering/Roadmap:

- 1st Gen: 24 hardware threads, scalar
- 2nd Gen: 16 hardware threads, scalar
- 3rd Gen: 16 hardware threads, 2 way superscalar

in production
in design
planning

up mem

SAFARI Live Seminar: Fabrice Devaux, 2 Feb 2022
Posted on January 15, 2022 by ewent

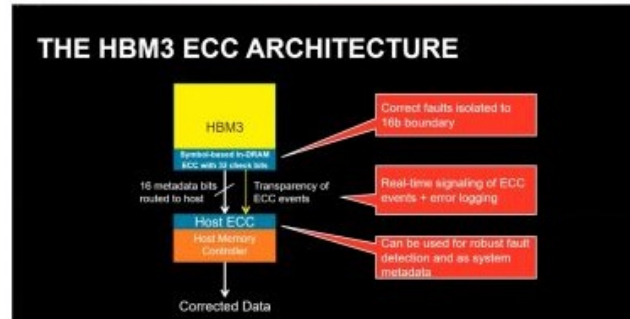
Join us for our joint SAFARI Live Seminar & EFCL Seminar with Fabrice Devaux, UPMEM
Wednesday, February 2 2022 at 11:00 am Zurich time (CET)

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=1

SAFARI Live Seminars (Upcoming Talks)

SAFARI Live Seminars in Computer Architecture

HBM3 RAS: The Journey to Enhancing Die-Stacked DRAM Resilience at Scale



SPEAKER
Sudhanva Gurumurthi
AMD Fellow



OCT 25, 2022 4:00PM CEST

SAFARI Live Seminar: Sudhanva Gurumurthi, Oct 25 2022

Posted on September 6, 2022 by ewent

We're excited to have **Sudhanva Gurumurthi** with us for our upcoming **SAFARI Live Seminar!**

Date: Tuesday, October 25 at 4:00 pm Zurich time (CEST)

Speaker: **Sudhanva Gurumurthi**, AMD Fellow

Link: Livestream on YouTube [Link](#)

Title: HBM3 RAS: The Journey to Enhancing Die-Stacked DRAM Resilience at Scale

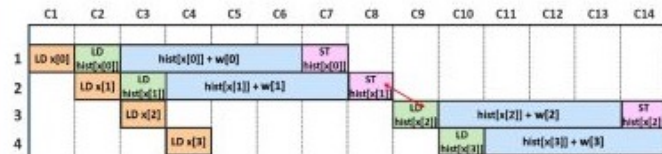
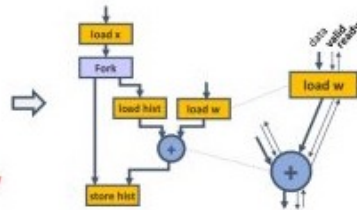
SAFARI Live Seminars (Upcoming Talks)

SAFARI Live Seminars in Computer Architecture

From C/C++ Code to High-Performance Dataflow Circuits

```
for (i=0; i<N; i++) {  
  hist[x[i]] = hist[x[i]] + w[i];  
}
```

```
1: x[0]=5 + ld hist[5]; st hist[5];  
2: x[1]=4 + ld hist[4]; st hist[4];  
3: x[2]=4 + ld hist[4]; st hist[4];  
    read-after-write dependency
```



SPEAKER
LANA JOSIPOVIĆ
Digital Systems and Design
Automation Group, ETH Zurich

NOV 7, 2022 4:00PM CST

SAFARI Live Seminar: Lana Josipović, Nov 7 2022

Posted on September 9, 2022 by ewent

Join us for our upcoming **SAFARI Live Seminar**

Date: Monday, November 7 at 4:00 pm Zurich time (CET)

Speaker: **Lana Josipovic**, DYNAMO Research Group, ETH Zurich

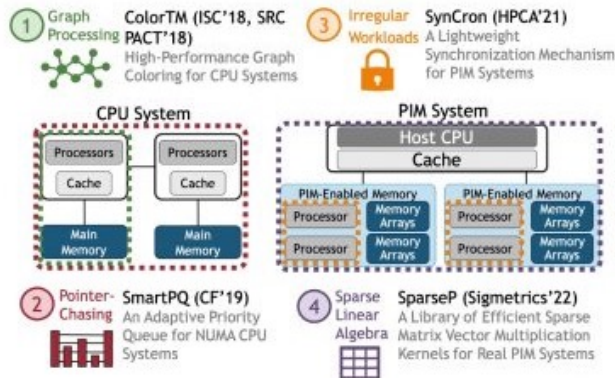
Link: Livestream on YouTube [Link](#)

Title: From C/C++ Code to High-Performance Dataflow Circuits

SAFARI Live Seminars (Upcoming Talks)

SAFARI Live Seminars in Computer Architecture

Accelerating Irregular Applications via Efficient Synchronization and Data Access Techniques



SPEAKER
Christina Giannoula
Computing Systems Lab, NTUA

ETH zürich

SAFARI
SAFARI Research Group



NOV 9, 2022 4:00PM CST

SAFARI Live Seminar, Christina Giannoula, Nov 9 2022

Posted on September 15, 2022 by ewent

Join us for our upcoming **SAFARI Live Seminar**

Date: Wednesday, November 9 at 4:00 pm Zurich time (CET)

Speaker: **Christina Giannoula**, School of Electrical and Computer Engineering, NTUA

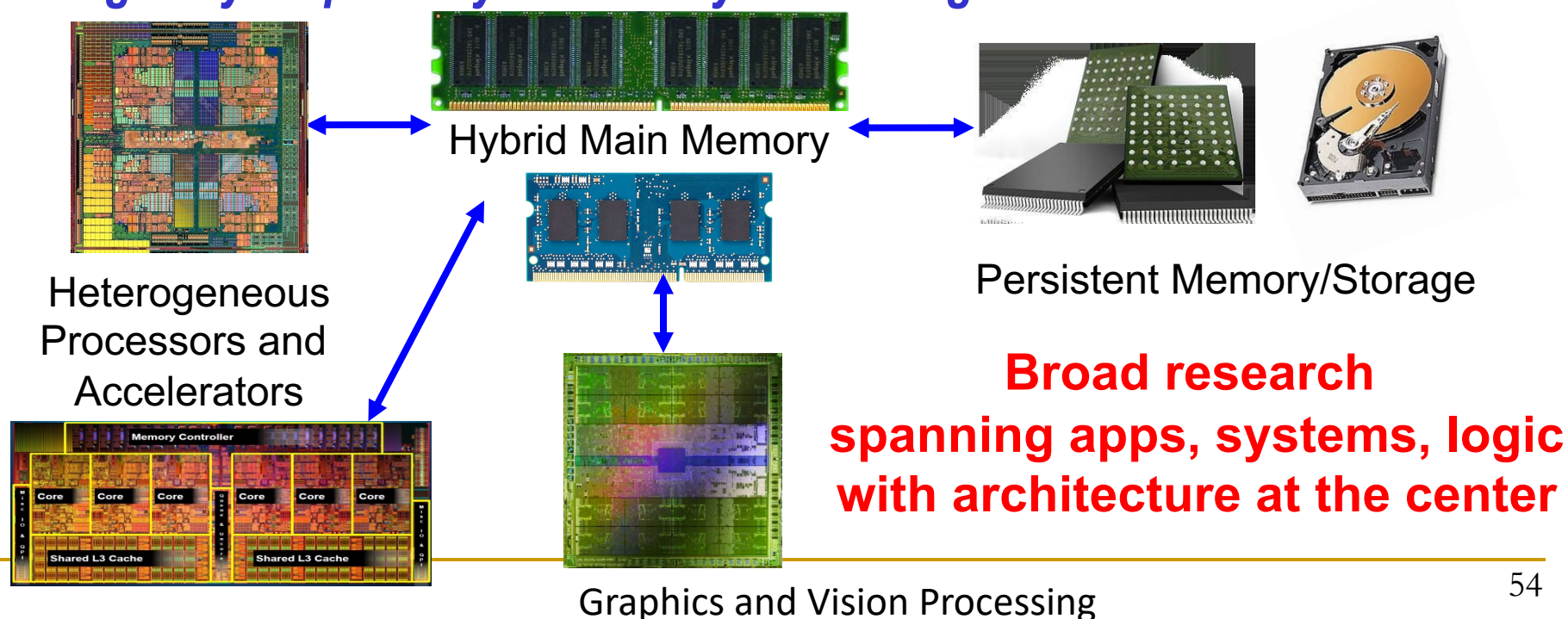
Link: Livestream on YouTube [Link](#)

Title: Accelerating Irregular Applications via Efficient Synchronization and Data Access Techniques

Current Research Focus Areas

Research Focus: Computer architecture, HW/SW, bioinformatics

- **Memory and storage (DRAM, flash, emerging), interconnects**
- **Heterogeneous & parallel systems, GPUs, systems for data analytics**
- **System/architecture interaction, new execution models, new interfaces**
- **Energy efficiency, fault tolerance, hardware security, performance**
- **Genome sequence analysis & assembly algorithms and architectures**
- **Biologically inspired systems & system design for bio/medicine**



Course Requirements and Expectations

- Attendance required for all meetings
- Study the learning materials
- Each student will carry out a hands-on project
 - Build, implement, code, and design with close engagement from the supervisors
- Participation
 - Ask questions, contribute thoughts/ideas
 - Read relevant papers

We will help in all projects!

If your work is really good, you may get it published!

Course Website

- https://safari.ethz.ch/projects_and_seminars/doku.php?id=heterogeneous_systems
- Useful information about the course
- Check your email frequently for announcements
- We also have Moodle for Q&A

Meeting 1

- Recommended materials:

1. An introduction to SIMD processors and GPUs (Dr. Juan Gomez Luna, lecture).

[\(PDF\)](#) [\(PPT\)](#) [Video](#)

2. An introduction to GPUs and heterogeneous programming (Dr. Juan Gomez Luna, lecture).

[\(PDF\)](#) [\(PPT\)](#) [Video](#)

- Other recommended materials:

3. Juan Gomez-Luna, Izzat El Hajj, Li-Wen Chang, Victor Garcia-Flores, Simon Garcia de Gonzalo, Thomas B. Jablin, Antonio J. Peña and Wen-mei Hwu,

["Chai: Collaborative Heterogeneous Applications for Integrated-architectures"](#)

Proceedings of the 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Santa Rosa, California, April 2017.

<https://chai-benchmarks.github.io> <https://github.com/chai-benchmarks/chai>

4. Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,

["NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"](#)

Proceedings of the [30th International Conference on Field-Programmable Logic and Applications \(FPL\)](#), Gothenburg, Sweden, September 2020.

[\[Slides \(pptx\) \(pdf\)\]](#)

[\[Lightning Talk Slides \(pptx\) \(pdf\)\]](#)

[\[Talk Video\]](#) (23 minutes)]

5. Mohammed Alser, Taha Shahroodi, Juan Gomez Luna, Can Alkan, and Onur Mutlu,

["SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs"](#)

[Bioinformatics](#), 26 December 2020.

[\[Source Code\]](#) [\[Online link at Bioinformatics Journal\]](#)

6. Real Processing-in-DRAM with UPMEM (Dr. Juan Gomez Luna, SAFARI Live Seminar, July 2021).

["Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture"](#)

Preprint in [arXiv](#), 9 May 2021.

[\[PrIM Benchmarks Source Code\]](#)

[\[Slides \(pptx\) \(pdf\)\]](#)

[\[SAFARI Live Seminar Slides \(pptx\) \(pdf\)\]](#)

[\[SAFARI Live Seminar Video\]](#) (2 hrs 57 mins)]

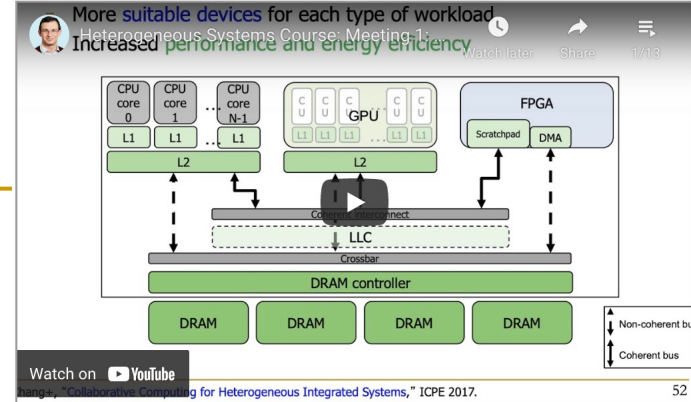
Meeting 2

- We will **announce the projects** and will give you some description about them
- We will give you a chance to select a project
- Then, we will have **1-1 meetings** to match your interests, skills, and background with a suitable project
- It is important that you **study the learning materials** before our next meeting!

Next Meetings

- Individual meetings with your mentor/s
- Tutorials and short talks
 - GPU programming
 - Recent research works
- Presentation of your work

Hetero. Systems (Spring 2022)



Spring 2022 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems

Youtube Livestream:

- https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm

Project course

- Taken by Bachelor's/Master's students
- GPU and Parallelism lectures
- Hands-on research exploration
- Many research readings

<https://www.youtube.com/onurmutlulectures>

Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	YouTube Premiere	M1: P&S Course Presentation PDF PPT	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	YouTube Premiere	M2: SIMD Processing and GPUs PDF PPT		
W3	29.03 Tue.	YouTube Premiere	M3: GPU Software Hierarchy PDF PPT		
W4	05.04 Tue.	YouTube Premiere	M4: GPU Memory Hierarchy PDF PPT		
W5	12.04 Tue.	YouTube Premiere	M5: GPU Performance Considerations PDF PPT		
W6	19.04 Tue.	YouTube Premiere	M6: Parallel Patterns: Reduction PDF PPT		
W7	26.04 Tue.	YouTube Premiere	M7: Parallel Patterns: Histogram PDF PPT		
W8	03.05 Tue.	YouTube Premiere	M8: Parallel Patterns: Convolution PDF PPT		
W9	10.05 Tue.	YouTube Premiere	M9: Parallel Patterns: Prefix Sum (Scan) PDF PPT		
W10	17.05 Tue.	YouTube Premiere	M10: Parallel Patterns: Sparse Matrices PDF PPT		
W11	24.05 Tue.	YouTube Premiere	M11: Parallel Patterns: Graph Search PDF PPT		
W12	01.06 Wed.	YouTube Premiere	M12: Parallel Patterns: Merge Sort PDF PPT		
W13	07.06 Tue.	YouTube Premiere	M13: Dynamic Parallelism PDF PPT		
W14	15.06 Wed.	YouTube Premiere	M14: Collaborative Computing PDF PPT		
W15	24.06 Fri.	YouTube Premiere	M15: GPU Acceleration of Genome Sequence Alignment PDF PPT		
W16	14.07 Thu.	YouTube Premiere	M16: Accelerating Agent-based Simulations PDF ODP		

Exploiting Data Parallelism: SIMD Processors and GPUs

Recall: Flynn's Taxonomy of Computers

- Mike Flynn, “**Very High-Speed Computing Systems**,” Proc. of IEEE, 1966
- **SISD**: Single instruction operates on single data element
- **SIMD**: Single instruction operates on multiple data elements
 - Array processor
 - Vector processor
- **MISD**: Multiple instructions operate on single data element
 - Closest form: systolic array processor, streaming processor
- **MIMD**: Multiple instructions operate on multiple data elements (multiple instruction streams)
 - Multiprocessor
 - Multithreaded processor

Recall: MMX Example: Image Overlaying (I)

- Goal: Overlay the human in image x on top of the background in image y

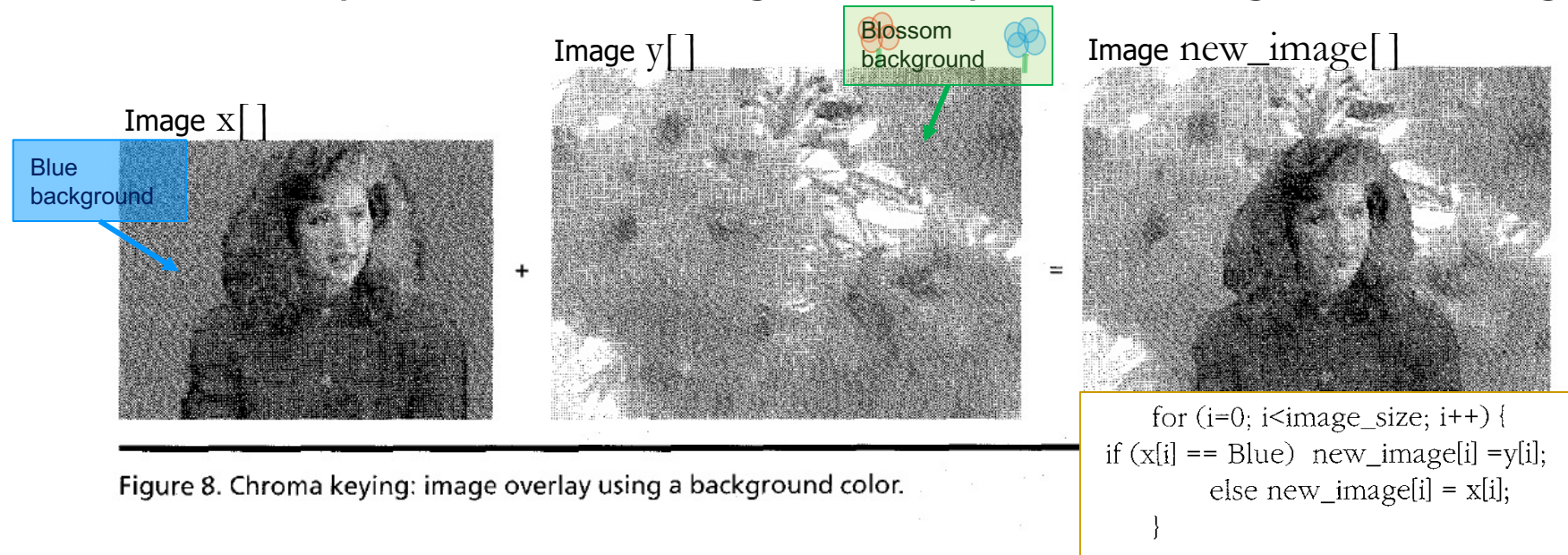


Figure 8. Chroma keying: image overlay using a background color.

PCMPEQB MM1, MM3

MM1	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
MM3	X7!=blue	X6!=blue	X5=blue	X4=blue	X3!=blue	X2!=blue	X1=blue	X0=blue
MM1	0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF



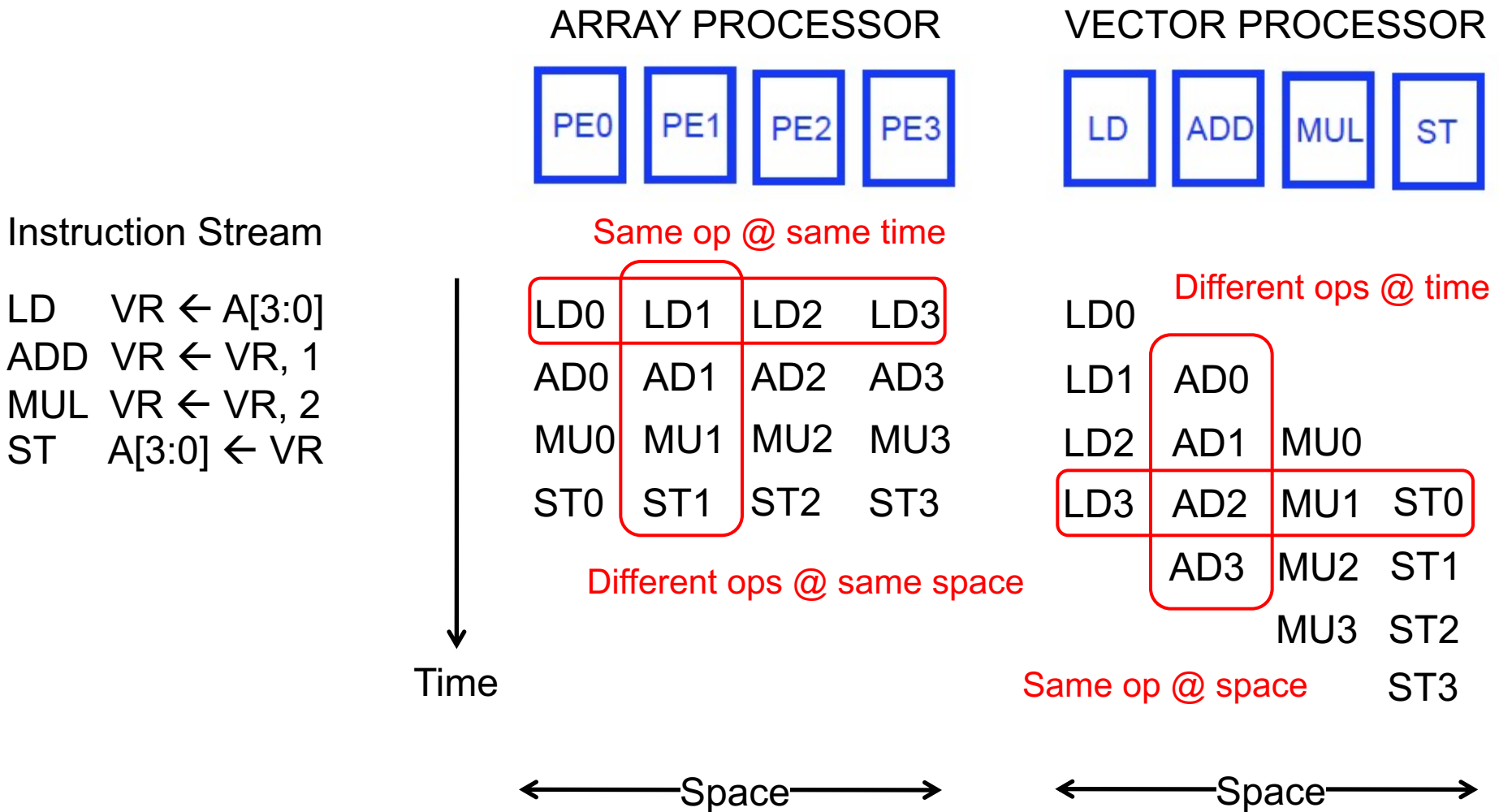
Bitmask

Figure 9. Generating the selection bit mask.

SIMD Processing

- Single instruction operates on multiple data elements
 - In time or in space
- Multiple processing elements (PEs), i.e., execution units
- Time-space duality
 - **Array processor**: Instruction operates on multiple data elements at the **same time** using **different spaces (PEs)**
 - **Vector processor**: Instruction operates on multiple data elements in **consecutive time steps** using the **same space (PE)**

Array vs. Vector Processors



NVIDIA A100 Core

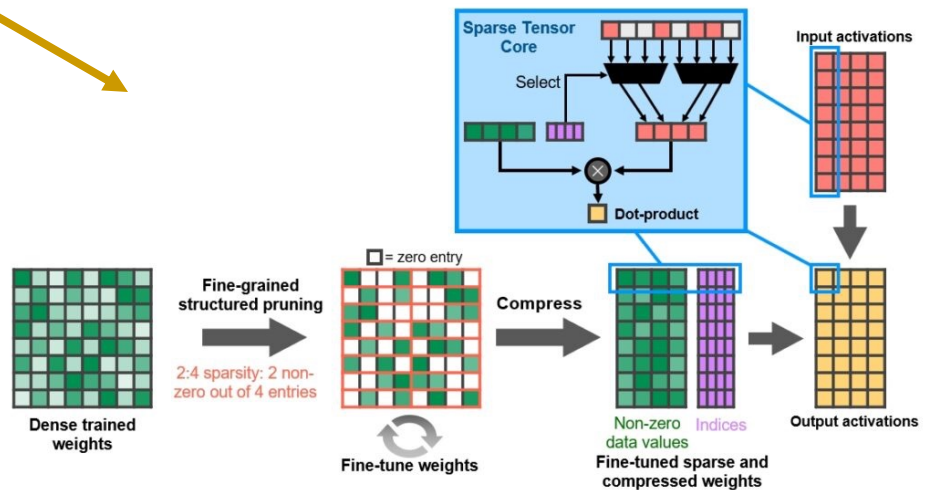


GPU compute throughput:

19.5 TFLOPS Single Precision

9.7 TFLOPS Double Precision

312 TFLOPS for Deep Learning (Tensor cores)



Vector Processor Disadvantages

- Works (only) if parallelism is regular (data/SIMD parallelism)
 - ++ Vector operations
 - Very inefficient if parallelism is irregular
 - How about searching for a key in a linked list?

To program a vector machine, the compiler or hand coder must make the data structures in the code fit nearly exactly the regular structure built into the hardware. That's hard to do in first place, and just as hard to change. One tweak, and the low-level code has to be rewritten by a very smart and dedicated programmer who knows the hardware and often the subtleties of the application area. Often the rewriting is

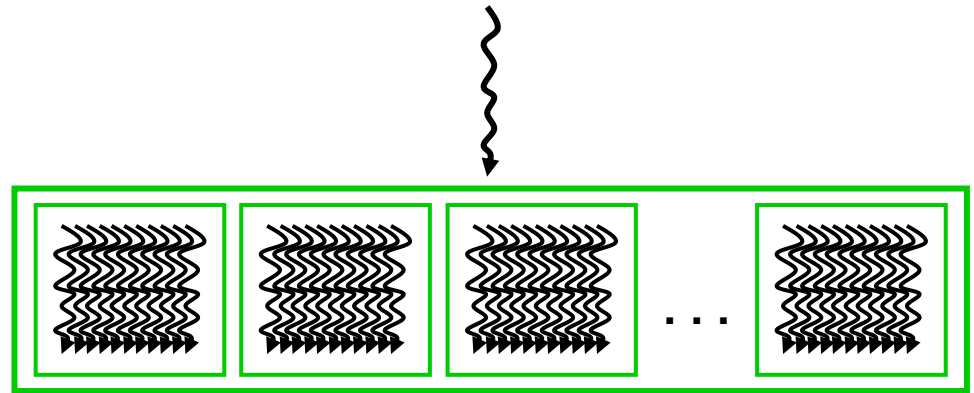
Warps *not* Exposed to GPU Programmers

- CPU threads and GPU kernels
 - Sequential or modestly parallel sections on CPU
 - Massively parallel sections on GPU: **Blocks of threads**

Serial Code (host)

Parallel Kernel (device)

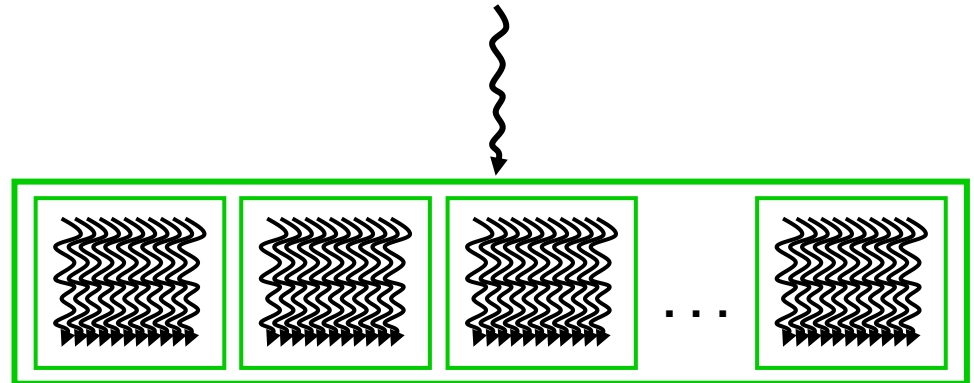
```
KernelA<<<nBlk, nThr>>>(args);
```



Serial Code (host)

Parallel Kernel (device)

```
KernelB<<<nBlk, nThr>>>(args);
```



Sample GPU SIMT Code (Simplified)

CPU code

```
for (ii = 0; ii < 100000; ++ii) {  
    C[ii] = A[ii] + B[ii];  
}
```



CUDA code

```
// there are 100000 threads  
__global__ void KernelFunction(...) {  
    int tid = blockDim.x * blockIdx.x + threadIdx.x;  
    int varA = aa[tid];  
    int varB = bb[tid];  
    C[tid] = varA + varB;  
}
```

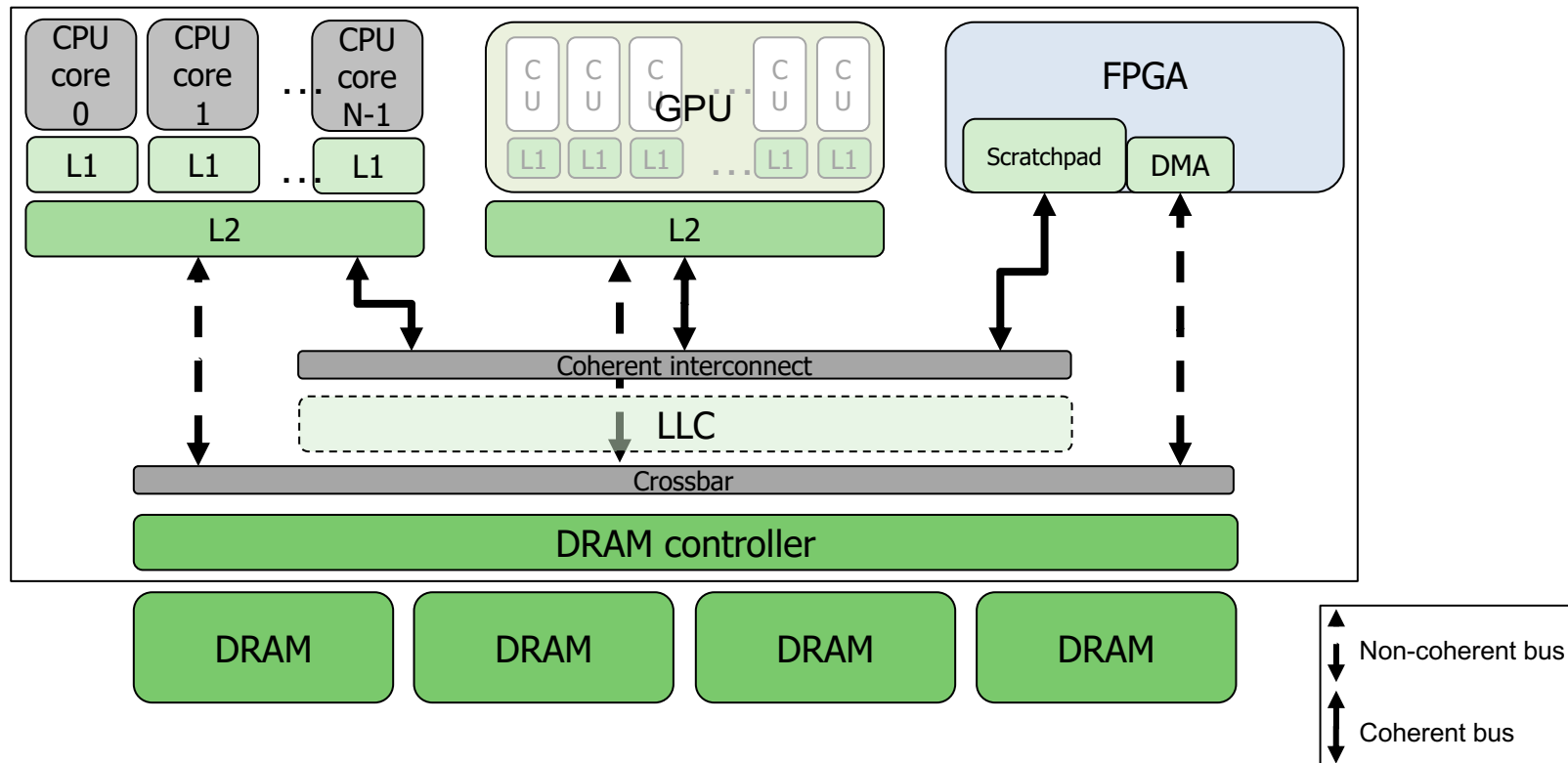

Vector Processor Disadvantages

- Works (only) if parallelism is regular (data/SIMD parallelism)
 - ++ Vector operations
 - Very inefficient if parallelism is irregular
 - How about searching for a key in a linked list?

To program a vector machine, the compiler or hand coder must make the data structures in the code fit nearly exactly the regular structure built into the hardware. That's hard to do in first place, and just as hard to change. One tweak, and the low-level code has to be rewritten by a very smart and dedicated programmer who knows the hardware and often the subtleties of the application area. Often the rewriting is

Heterogeneous Computing Systems

- The end of Moore's law created **the need for heterogeneous systems**
 - More **suitable devices** for each type of workload
 - Increased **performance and energy efficiency**



Chai Benchmark Suite

- Heterogeneous execution on CPU, GPU, FPGA
- Collaboration patterns
 - 8 data partitioning benchmarks
 - 3 coarse-grain task partitioning benchmarks
 - 3 fine-grain task partitioning benchmarks
- Discrete (D) and Unified (U) versions
- Chai versions
 - CUDA and OpenCL for CPU+GPU
 - OpenCL for CPU+FPGA
 - CUDA-Sim for Gem5-GPU



<https://chai-benchmarks.github.io>

P&S Heterogeneous Systems

Programming Heterogeneous Computing
Systems with GPUs and other Accelerators

Dr. Juan Gómez Luna

Prof. Onur Mutlu

ETH Zürich

Fall 2022

3 October 2022